MCN: 0080262-01
Release 5.0

COPPEREDGE™

Installation and Operating Guide

CopperEdge 150 DSL Concentrator

COPPER MOUNTAIN®

Part Number 0080262-01
May 2001
Release 5.0

San Diego Facility
10145 Pacific Heights Boulevard, Suite 100
San Diego, California 92121

Palo Alto Facility
2470 Embarcadero Way
Palo Alto, California

Limited Warranty

Copper Mountain Networks, Inc. ("Copper Mountain") warrants the CopperEdge™ 150 to be free from defects in materials or workmanship for a period of one (1) year from the date of shipment from Copper Mountain's factory. Should your CE150 fail during the warranty period, Copper Mountain will, at its option and as its sole and exclusive obligation under this warranty, repair or replace it with a like product, which may include new or refurbished parts or components. This warranty is extended only to the original purchaser and only covers failures due to defects in materials and workmanship which occur during normal use during the period of the warranty. It does not cover damage which occurs in shipment or failures resulting from misuse, negligence, accident, improper storage, installation or testing, unusual electrical stress, fire, lightning, other environmental hazards, unauthorized attempts at repair, operation inconsistent with published electrical and environmental specifications, or if the Product was maintained in a manner other than described in this document, or if the serial number or other identifications markings have been altered, removed or rendered illegible. Expendable components such as batteries or cabling external to the unit are not covered by this warranty.

For specific terms and conditions of the product warranty and Copper Mountain's obligations thereto, please refer to the warranty section of your purchase agreement.

In order to exercise your rights to repairs under this warranty, you must first contact Copper Mountain at 888-267-7374 (for calls made from U.S. locations only) or 858-410-7100 (for all calls made from outside the U.S.) to obtain authorization (including tracking number) and instructions for return of the product(s).

EXCEPT FOR THE EXPRESS WARRANTIES SET FORTH HEREIN AND IN THE APPLICABLE PURCHASE AGREEMENT, COPPER MOUNTAIN MAKES NO WARRANTIES, WHETHER EXPRESS, IMPLIED OR STATUTORY AND HEREBY EXPRESSLY DISCLAIMS ANY AND ALL SUCH WARRANTIES, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WITH RESPECT TO THE PRODUCTS PURCHASED. NO REPRESENTATION OR OTHER AFFIRMATION OF FACT, WHETHER MADE BY COPPER MOUNTAIN OR OTHERWISE, SHALL BE DEEMED TO BE A WARRANTY FOR ANY PURPOSE OR GIVE RISE TO ANY LIABILITY OF COPPER MOUNTAIN.

The laws in some localities do no allow limitations on the warranty period so the above exclusions or limitations may not be applicable. This warranty provides specific legal rights. You may have other rights with respect to this product, depending on local laws.

Software License Agreement

Use of this Product is subject to the Software License Grant contained in the purchase agreement used to originally purchase this Product from Copper Mountain.

Except as expressly permitted by applicable law, Customer or any third party to which the software may be licensed as permitted under the original purchase agreement, may not alter, modify, adapt or translate, create any derivative works based on the software, reverse engineer, decompile, disassemble or otherwise attempt to derive the human-readable source code for the software licensed with this Product.

* * * * *

Should you have questions regarding the above Product warranty or Software License Grant, please contact Copper Mountain Networks, Contracts Department at 858-410-7277.

Safety Notices

Batteries

The CE150 is equipped with a lithium battery to maintain certain system parameters that you set in the installation and configuration process.

There is a danger of explosion if the battery is replaced incorrectly. Replace only with the same or equivalent type recommended by the manufacturer. Dispose of used batteries in accordance with the manufacturer's instructions.

Fuses

The telephone line interface module (DSL module) and AC Power module in the CE150 are equipped with over-current protection devices.

For continued protection against the risk of fire and electric shock, make sure to replace any fuse with a fuse of the same type and rating as the original.

Protective Earthing

In some cases, the CE150 needs additional grounding to be in compliance with local Safety Standards.

If the CE150 is installed such that the SDSL loops or analog modem interfaces are connected to telephone wiring that is external to the building, the unit must be permanently connected to an Earth Ground in accordance with local and National Safety Standards, using the protective earthing terminal on the rear of the equipment. The Earth Ground must be professionally installed by a qualified electrician. Refer to instructions provided in Chapter 2.

Installation and Maintenance

The CE150 must be installed in a restricted access area, which is only accessible to trained and qualified personnel.

All CE150 installation and maintenance must be performed by trained and qualified personnel.

North America Regulatory Notices

United States, FCC Part 15 Emissions Notice

The Copper Mountain CE 150 DSL Concentrator has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules (Federal Communications Commission).

Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

The FCC Class A limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user will be required to correct the interference at his own expense.

The manufacturer is not responsible for any radio or television interference caused by using other than recommended cables or by unauthorized changes or modifications to this equipment. Unauthorized changes or modifications could void the user's authority to operate this equipment.

Industry Canada, ICES-003 Emissions Notice

The Copper Mountain CE 150 DSL Concentrator is a Class A digital apparatus that complies with Canadian ICES-003.

United States, FCC Part 68 Telecommunications Notice

The Copper Mountain CE 150 DSL Concentrator Modem complies with Part 68 of the FCC rules. On the backside of the CE 150 is a label that contains, among other information, the FCC registration number and Ringer Equivalence Number (REN) for this equipment. If requested, this information must be provided to the Telephone Company.

The REN is used to determine the quantity of devices that may be connected to the telephone line. Excessive RENs on the telephone line may result in the device not ringing in response to an incoming call. In most, but not all, areas the sum of the RENs should not exceed five (5.0). To be certain of the number of devices that may be connected to the line as determined by the REN, contact the telephone company to determine the maximum REN for the calling area.

This equipment cannot be used on telephone company-provided coin service. Connection to party lines is subject to state tariffs.

This equipment uses the following Universal Service Order Code (USOC) jack: RJ-11C.

This equipment is designed to be connected to the telephone network or premises wiring using a compatible jack that is FCC Part 68 compliant. An FCC compliant telephone cord and modular plug is provided with this equipment.

If this equipment causes harm to the telephone network, the telephone company will notify you in advance that temporary discontinuance of service may be required. If advance notice is not practical, the telephone company will notify the customer as soon as possible. Also, you will be advised of your right to file a complaint with the FCC if you believe it is necessary.

The telephone company may make changes to its facilities, equipment, operations, or procedures that could affect the operation of the equipment. If this happens, the telephone company will provide advance notice in order for you to make necessary modification in order to maintain uninterrupted service.

If trouble is experienced with this equipment, please contact Copper Mountain Networks, Inc., at (888) 611-4266 for repair and/or warranty information. If the trouble is causing harm to the telephone network, the telephone company may request you to disconnect the equipment from the network until the problem is resolved.

It is recommended that the customer install an AC surge arrestor in the AC outlet to which this device is connected. This is to avoid damage to the equipment caused by local lightning strikes and other electrical surges.

Canadian, Industry Canada CS-03 Telecommunications Notice

NOTICE: The Industry Canada label identifies certified equipment. This certification means that the equipment meets certain telecommunications network protective operational and safety requirements in the appropriate Terminal Equipment Technical Requirements documents. The Department does not guarantee the equipment will operate to the user's satisfaction.

Before installing this equipment, users should ensure that it is permissible to be connected to the facilities of the local telecommunications company. The equipment must also be installed using an acceptable method of connection. The customer should be aware that compliance with the above conditions may not prevent degradation of service in some situations.

Repairs to certified equipment should be coordinated by a representative designated by the supplier. Any repairs or alterations made by the user to this equipment, or equipment malfunctions, may give the telecommunications company cause to request the user to disconnect the equipment.

Users should ensure for their own protection that the electrical ground connections of the power utility, telephone lines, and internal metallic water pipe system, if present, are connected together. This precaution may be particularly important in rural areas.

C A U T I O N

Users should not attempt to make such connections themselves, but should contact the appropriate electric inspection authority, or electrician, as appropriate.

The Ringer Equivalence Number (REN) of this device is 0.6.

The Ringer Equivalence Number (REN) assigned to each terminal device provides an indication of the maximum number of terminals allowed to be connected to a telephone interface. The termination of an interface may consist of any combination of devices subject only to the requirement that the sum of the Ringer Equivalence Numbers of all the devices does not exceed 5.

This equipment uses Canadian CA11A Jacks.

Table of Contents

List of Illustrations

List of Tables

Revision History

Rev	Rev Date	Summary of Changes
01	February 2001	Limited Availability release.
02	May 2001	General Availability release.

System Software and Applicability

This document applies to CopperEdge 200 systems delivered under Hardware/Software Release 4.0.

Document Conventions

The following conventions are used throughout this document:

`this font`	Indicates user input, such as configuration commands.
`This font`	Indicates output from the CopperCraft command line interface or other sources.
This font	Emphasizes new terms, or the names of other sections, chapters, and books.
Examples of configuration commands:	`CRAFT>` **`set cmiface [1.32.1.44] netmodel=ip farendaddr=192.168.99.2`** Examples show link- or user-specific information such as IP addresses, subnet masks, or MAC addresses. Unless otherwise specified, all such data is fictitious and is provided for illustrative purposes only.

N O T E

Additional information pertaining to the topic.

C A U T I O N

Information alerting you to a situation that could result in damage to the data or system.

C A U T I O N

Information alerting you to a situation where electrostatic discharge could damage the data or system.

W A R N I N G

Information alerting you to a situation that could result in personal injury.

Chapter 1
Introducing the CopperEdge 150

The CopperEdge™ 150 (CE150) DSL Concentrator is at the heart of the Copper Mountain high speed DSL system for occupants of multiple tenant units, such as office and professional buildings. Each CE150 unit can distribute DSL data packets at high speed to as many as 48 subscriber lines over two-conductor twisted-pair copper wire.

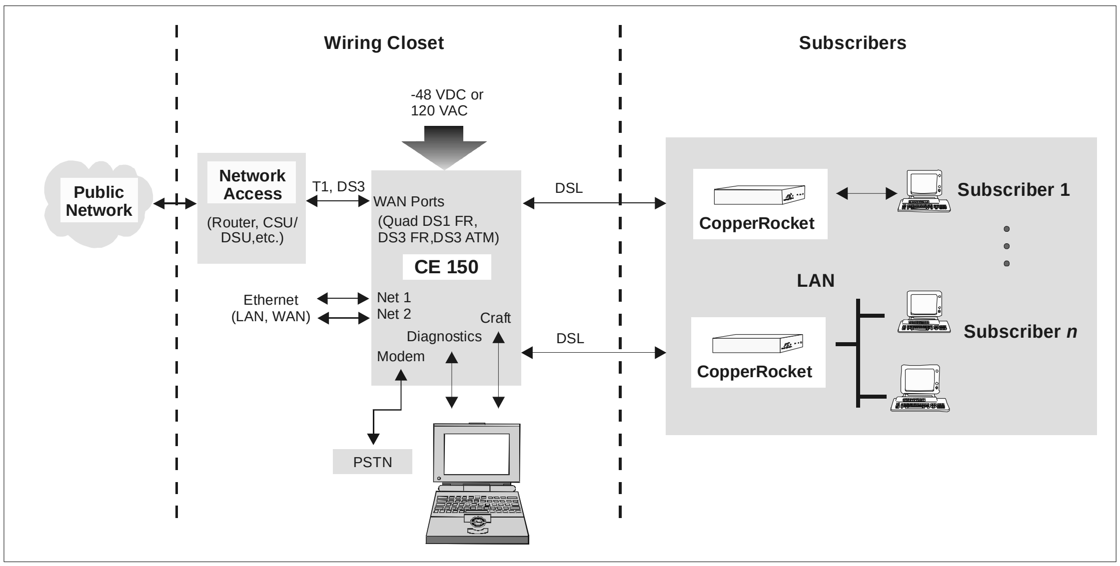
A high-speed DSL access device, such as Copper Mountain's CopperRocket™ 201 SDSL or CopperRocket 202 SDSL IMUX or CopperRocket 508 Integrated Access Device, seamlessly connects the subscriber's PC or LAN through the DSL physical link to the CE150 at a central location.

Concept of Operation

The following illustration shows a typical deployment of the elements of the Copper Mountain DSL system.

At the subscriber site, the CopperRocket (also referred to as the Customer Premise Equipment or CPE) connects to the subscriber's LAN, PC, or other device with 10BASE-T Ethernet.

The CE150 receives data from each DSL link (subscriber line) at a central location in the office building. It adds packet header information to identify each of its connected subscribers, and aggregates the subscriber lines for transmission over the network.



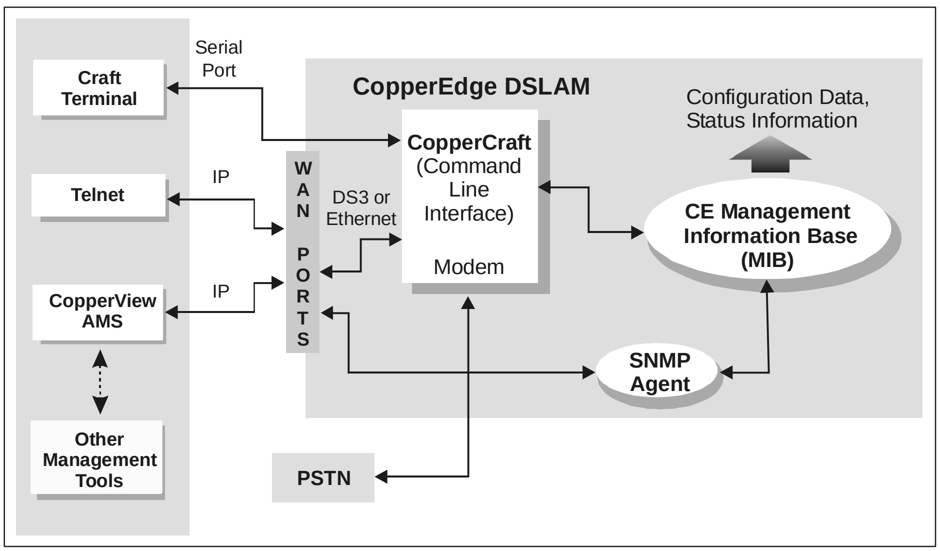
Status/Configuration Data Flow

Management Tools

Software to configure and manage the system consists of the CopperCraft™ command line/user interface. Remote monitoring and control through Telnet is also available.

A graphic user interface is available through the Element Manager software component of the CopperView™ DSL Access Management System (AMS). The CopperView AMS facilitates remote control and management of multiple CopperEdge concentrators simultaneously through SNMP. The CE150's software tools may be used individually or together to configure the CE150 and each of its links and to monitor status and performance.

As shown in the following illustration, the CopperCraft interface provides an intelligible, text-based mechanism by which an operator can directly monitor and control the CE150. When controlled by the CopperView Element Manager or other network management systems, the CE150's SNMP agent responds to commands originated by another computer (SNMP Manager).



Concept of Operation

Third-Party Customer Premise Equipment (CPE)

As a leader in the move toward standardization in the xDSL industry, Copper Mountain has instituted an ongoing program to maximize interoperability of the CE150 with the DSL customer premise equipment (CPE) of other manufacturers, as well as with its own CopperRocket product line.

Many CPEs from third-party manufacturers have been certified for use with the . Because the catalog of certified products is continually growing, check with your Copper Mountain sales representative for an up-to-the-minute list of compatible CPE products.

Hardware Features

Specifications

All specifications are subject to change without notice. New modules are being added all the time. In addition, special releases occur; supplemental documentation describes those modules and their releases. This documentation is later incorporated in the standard operating guides.

DSL Platform

- Flexible, modular design supports current and future DSL interfaces, transmission, and backbone network technologies
- Seven-slot chassis; up to two DSL modules (maximum of 48 physical ports), in any combination of SDSL, ADSL (G.dmt or G.lite), or T1 per CE150 shelf

Physical Dimensions

- Size: 6.97 inches H x 19 inches W x 10.75 inches D (17.7 centimeters x 48.3 centimeters x 27.3 centimeters)
- Weight (approximate) with full circuit-module complement: 33 lbs (14.85 kg)
- Rack mounting options: EIA 19 inches (48.3 centimeters) or 23 inches (58.4 centimeters)
- Wall mounting option

Input Power

One of the following options:

- AC Power module:
  - ⇒ 120 VAC, 5 A, 60 Hz
  - ⇒ A Blank Deflection Shield must be installed in slot 6
  - ⇒ The AC Power Module must be installed in slot 7
- DC Power module:
  - ⇒ -48 VDC, 8 A
  - ⇒ A blank face panel must be installed in slot 6 or 7
  - ⇒ The DC Power Module must be installed in slot 6 or 7

Operating Environment

- Temperature: 32° to 122°F (0° to 50°C)
- Humidity: 10 to 80%, non-condensing
- Altitude: to 10,000 feet

System Control Modules

- SCM-1 and SCM-2:
  - ⇒ Intel Pentium processor (P54C)
  - ⇒ Up to 200 MHz internal clock, 66 MHz front side bus (FSB)
  - ⇒ Up to 512 Kbytes of secondary cache (L2 cache)
  - ⇒ 32 Mbytes DRAM, 66 MHz
  - ⇒ 8 MBytes of on-board flash memory
  - ⇒ One 10Base-T/100Base-TX Ethernet port with an RJ-45 connector
  - ⇒ Two DB9 male serial ports (Craft and Diagnostic) with EIA/TIA-232 Serial Interface connectors
  - ⇒ 33,600 Kbps analog modem with error correction and data compression support
- SCM-3:
  - ⇒ Intel Pentium III processor (EMC2)
  - ⇒ 500 MHz internal clock, 100 MHz front side bus (FSB)
  - ⇒ 256 Kbytes of secondary cache (L2 cache)
  - ⇒ 128 Mbytes DRAM, 100 MHz
  - ⇒ 64 MBytes of on-board flash memory through a CompactFlash type 1 module
  - ⇒ Two 10Base-T/100Base-TX Ethernet ports with RJ-45 connectors (the second port is not supported in this release)
  - ⇒ Two 16C550 PC-Compatible serial ports (Craft and Diagnostic) with RJ-45 connectors
  - ⇒ 33,600 Kbps analog modem with error correction and data compression support
  - ⇒ Pinhole reset switch
  - ⇒ For optimal system performance, we recommend you use the Buffer Control Module 2 with the SCM-3

Buffer Control Modules

- BCM-1:
  - ⇒ i960CF (33 MHz) processor
  - ⇒ 4 Kbytes of internal instruction cache
  - ⇒ 1 Kbyte of internal data cache (direct-mapped, write-through)
  - ⇒ 1 Kbyte internal SRAM
  - ⇒ Can be used with SCM-1, SCM-2, or SCM-3
- BCM-2:
  - ⇒ Clock-doubled i960HD (33/66 MHz) processor
  - ⇒ 16 Kbytes of internal instruction cache (four-way set associative)
  - ⇒ 8 Kbytes of internal data cache (four-way set associative)
  - ⇒ 2 Kbytes internal RAM
  - ⇒ Can only be used with SCM-3

Line Modules

- SDSL 30X module (24 physical ports): multi-speed 1.568 Mbps, 1.040 Mbps; and 784, 416, 320, 208, and 160 Kbps 2B1Q
- ADSL G.lite module (24 physical ports): multi-speed 64 Kbps to 2.336 Mbps
- ADSL G.dmt module (24 physical ports): multi-speed 64 Kbps to 6.144 Mbps
- T1 Line Module (12 physical ports): 1.544 Mbps $\pm$ 50 bps
- Input/Output Impedance at all DSL ports: Balanced, 135 Ω

WAN Modules

- Two-port Universal Serial I/O—EIA-449 (CCITT V.36), X.21
- One-port DS3 (45 Mbps) Frame Relay, dual (Tx/Rx) 75-Ohm Coaxial
- One-port DS3 (45 Mbps) ATM, dual (Tx/Rx) 75-Ohm Coaxial
- Four-port DS1 (T1 PMC), 100-Ohm unshielded twisted pair, using modular RJ-48C connectors
- Up to 2048 WAN VCs can be supported per CE150

Software Features

DSL Link Protocols

- Internet RFC-1483 Multiprotocol Encapsulation over Frame-Based User-to-Network Interface (FUNI)
- DSL ports also support RFC-1490, Q.922, Q.922-1490, 1973, 2364, HDLC, and HDLC PPP as implemented in various third-party CPEs

Packet Multiplexing Protocols

- TCP/IP
  - ⇒ RFC-791 Internet Protocol (IP)
  - ⇒ RFC-792 Internet Control Message Protocol (ICMP)
  - ⇒ RFC-768 User Datagram Protocol (UDP)
  - ⇒ RFC-826 Ethernet Address Resolution Protocol (ARP)
  - ⇒ RFC-1058 Gateway and Hosting Protocol
- Virtual Wide Area Net (VWAN) multiplexing and CopperVPN IP multiplexing for Copper Mountain's CopperRocket customer-premise equipment
- Frame multiplexing for third-party CPEs
- Policy-based packet multiplexing
- Security filtering using source or destination address or port

Network Standard Protocols

- ANSI T1.606 Frame Relay Architectural Framework
- ANSI T1.606 Addendum 1 for Congestion Management – CIR/BIR/BECN/FECN support
- ANSI T1.618 Core Aspects of Frame Protocol for Frame Relay
- Multiple standards of LMI management protocols including T1.617 Annex D, Q.933 Annex A, and LMI Rev 1.0a
- Frame Relay Forum UNI FRF.1
- Frame Relay NNI (FRF.2) supported on WAN links
- RFC-1157 SNMP Protocol
- RFC-1483 Multiprotocol Encapsulation over ATM
- RFC-1490 Multiprotocol Encapsulation over Frame Relay
- RFC-1973 PPP over Frame Relay
- RFC-2364 PPP over ATM
- Inverse ARP over Frame Relay (per RFC-2390)
- Inverse ARP over ATM (per RFC-2225)

Network Management

- Full SNMP functionality
- Internet RFC-1213 MIB-II, RFC-1315 Frame Relay MIB, RFC-1573 DS3/ES MIB, and Copper Mountain Proprietary MIB
- Proxy management for Copper Mountain CPEs
- EIA-2332D Console port/Craft interface
- Telnet (password protected) to CopperCraft CLI
- CopperCraft interface accepts up to four simultaneous management sessions through IP (Telnet and FTP), up to five sessions through SNMP, and one through the Craft serial port
- Ping utility (both originate and respond)
- CopperView DSL Access Management System with graphics-based element manager, a self-contained SNMP Manager
- Radius authentication and accounting for system-level operators, including those on the CE150

Reliability/Serviceability

- Hot-swap supported for all DSL modules; front-panel status indicators to aid fault isolation
- Optional power redundancy through dual load-sharing power supplies with automatic failover and fault monitoring
- Software and configuration downloads from flash memory or from external server through FTP
- Extensive performance monitoring through system-, trunk-, and port-level statistics
- Event and Alarm log, and extensive event trapping capability provide status information to NMS/SNMP Managers; front panel module shows alarm status, and network of contact closures enable audible and visible CO alarm devices

Product Certification in North America

- Product Safety:
  - ⇒ United States: UL 1950 3rd Edition, Safety of Information Technology Equipment, Including Electrical Business Equipment. UL Recognized Component
  - ⇒ Canada: CAN/CSA C22.2 No. 950-95 3rd Edition. CUL Recognized Component
- Product Emissions:
  - ⇒ United States: Compliant with the Code of Federal Regulations Title 47, Part 15, Radio Frequency Devices, Subpart B, Class A
  - ⇒ Canada: Compliant with Canadian ICES-003, Class A
- Telecommunications:
  - ⇒ United States: Compliant with the Code of Federal Regulations Title 47, Part 68, Connection of Terminal Equipment to Telephone Network
  - ⇒ Canada: Compliant with Canadian Certification Specification for terminal equipment, Terminal Systems, Network Protection Devices, Connection Arrangements and Hearing Aids Compatibility, CS-03

Release Notes

For complete information about new features, hardware/software compatibility, and late-breaking operational or other issues that the operator or installer should know about, consult the Release Notes document corresponding to the version of hardware and installed software in this CE150. A copy of the applicable Release Notes document should be packed with the CE150 in the same accessory pack as this manual.

In addition, electronic copies of all release notes are available at the Copper Mountain support FTP site. From your web browser, enter:

ftp://userid@support.coppermountain.com

Enter your password when prompted to complete your login.

Technical Support

Expert help with your Copper Mountain equipment and software is available through our Customer Service Center (CSC). Toll-free telephone support is available Monday through Friday from 6 a.m. to 6 p.m., Pacific time, at 888-611-4CMN (888-611-4266).

Optional service agreements are also available for those requiring extended support (24-hour, seven-day) and other premium services. Contact your Copper Mountain sales representative for details.

For assistance or information via e-mail, contact:

support@coppermountain.com

Technical information about DSL and Copper Mountain products and technology is also available at our web site at:

http://www.coppermountain.com

Chapter 2
Installation

This chapter describes the physical installation of the CE150 and provides information needed to establish its various physical connections.

The CE150 should only be installed by professionally trained personnel in restricted access equipment rooms, to prevent accidental injury to non-qualified personnel or damage to the unit. Complete information about the physical installation and turn-up is in the *CopperEdge Installer's Guide*.

Site Preparation

Before installing the CE150, read this section thoroughly.

Environmental Conditions

The only environmental conditions that may restrict use of the CE150 are temperature, humidity, and altitude:

- Temperature: 32° to 122° F (0° to 50° C)
- Humidity: 10 to 80%, non-condensing
- Altitude: to 10,000 feet

Required Services

Typically, the Telco will run the service cables to the edge of the Central Office cage. Then the Telco or another installer will install the proper patch panels at the edge of the cage.

Before you install the CE150, ensure that the following services are, or will be, in place.

- Telco connection
- DS3 ATM/Frame Relay service
- DS1 service
- LAN access
- A telephone line for dial-up access
- A regulated source of -48 VDC or 120 VAC Mains power

Required Equipment

System Hardware

The following hardware items are the *minimum* required for a full-function DSL access link. This minimum configuration will support 1 to 24 DSL ports.

- One CE150 chassis
- One System Control Module/WAN Module with a 10/100 Base-T Ethernet WAN connection
- One 24-port DSL module: SDSL or ADSL (G.Lite or G.dmt), or one 12-port T1 module

  The CE150 chassis can accommodate up to two DSL modules (SDSL ADSL, or T1 in any combination).
- One Buffer Control Module
- One Power Module. For NEBS compliance, you must and one blank panel or filler panel
- One Fan Tray Assembly

Your CE150 System Control Module may optionally include one of the following WAN connections:

- DS1 Quad-port-T1 Frame WAN
- DS-3 ATM WAN
- DS-3 Frame WAN

1	Buffer Control Module	Fan Assembly
2	WAN Module	
3	System Control Module with optional WAN Connection	
4	DSL Module	
5	DSL Module	
6	Blank Filler Panel	
7	AC or DC Power Module	

CE150 Module Arrangement

Control Console

A Control Console (a notebook or other portable PC) is typically used at installation and can be used by an on-site operator at any time to provide direct, serial-port access to the CopperCraft command line interface (CLI), to act as an external file server, or to monitor information at the CE150's Diagnostic port.

Cables

The following cables are included with your base system:

- Power Cable—stranded red, black, and green :
  - For connection to 120 VAC Mains Power: an IEC 320 AC Power Cable.
  - For connection to -48 VDC Power and Ground: #12 Stranded Red, Black, and Green or Green/Yellow Stripe UL Recognized Copper Wire; Closed #12 Wire Crimp Lugs.

N O T E

The CE150 can be powered by 120 VAC or 048 VDC, but not both.

- One DSL Connector Cable—10 feet of 24/25 pair cable with Amphenol connectors and locking clips. Connects the CopperEdge chassis to the Intermediate Distribution Frame. One cable is required for every 24 DSL ports.
- One 10Base-T Ethernet Cable—10 feet of Category 5 shielded twisted pair (STP) 10Base-T cable with RJ-45 connectors on each end (EIA-568A). Connects the CopperEdge Ethernet port to the LAN concentrator.
- One Local Craft Interface Cable—If your system is equipped with either a SCM1 or SCM2, it will need a EIA232 NUL Modem cable with 9-pin D-subminiature female connector to 9-pin D-subminiature male connector. It connects the serial port of a local terminal to the Craft or Diagnostic port on the System Control Module front panel.

You are responsible for providing these additional cables:

- Network connection cables:
  - For each DS3 WAN (Frame or ATM) module—two shielded 75-Ohm coax cables (RG-59/U) with BNC connectors on both ends. Connects the CopperEdge DS3 WAN module to the WAN concentrator.
  - For each DS1 WAN (Frame) module—Category 5 cabling with RJ-45 connectors on both ends. Connects the CopperEdge DS1 WAN module to the WAN concentrator.
- For each additional DSL line module, the appropriate DSL cable.
- For the dial-up modem, telephone cabling with RJ-11 connectors on both ends.

Tools

You will need the following items to install the CE150:

- Flat blade screwdriver
- No. 2 Phillips screwdriver (preferably with a magnetized tip)
- Voltmeter
- ESD static strap
- For connection of the -48 VDC Power and Ground, a #12 Wire-Lug Crimp tool
- Cable ties

Optional Equipment

For connection to the DIAG or Craft ports, you may need one or more of these items:

- Two serial cables (DB9), male-to-female
- One serial A/B switch
- One Gender changer, male-to-male, DB9

For access to DS3 ATM or Frame Relay Transmit/Receive monitor jacks, you will need two 75-Ohm double-braid shielded cables (RG316/U), with SMB connectors at both ends of each cable.

In North America, for dial-in access to CopperCraft (not available in the European Union), you will need to provide a serial modem cable (DB9/DB25).

System Software

In addition to the core data processing software, your CE150 software includes the CopperCraft CLI, SNMP agent, an FTP file server. The software will support the available CopperView AMS DSL Access Management System and CopperView EM Element Manager. Each of these tools may be employed in managing the CE150 and its links.

Configuration data is stored in the CE150's on-board flash memory. The CE150's embedded software, and the software for its CopperRocket CPEs, is upgradable by download from a remote host, allowing you to extend the useful life of your subscribers' CPEs.

CE150 Installation

C A U T I O N

All CE150 modules contain static-sensitive devices. If you must remove or handle modules for any reason, observe standard ESD precautions (ground straps for personnel and equipment, etc.). If you are unsure of the necessary precautions, contact Copper Mountain Technical Support for assistance.

Unpack the CE150

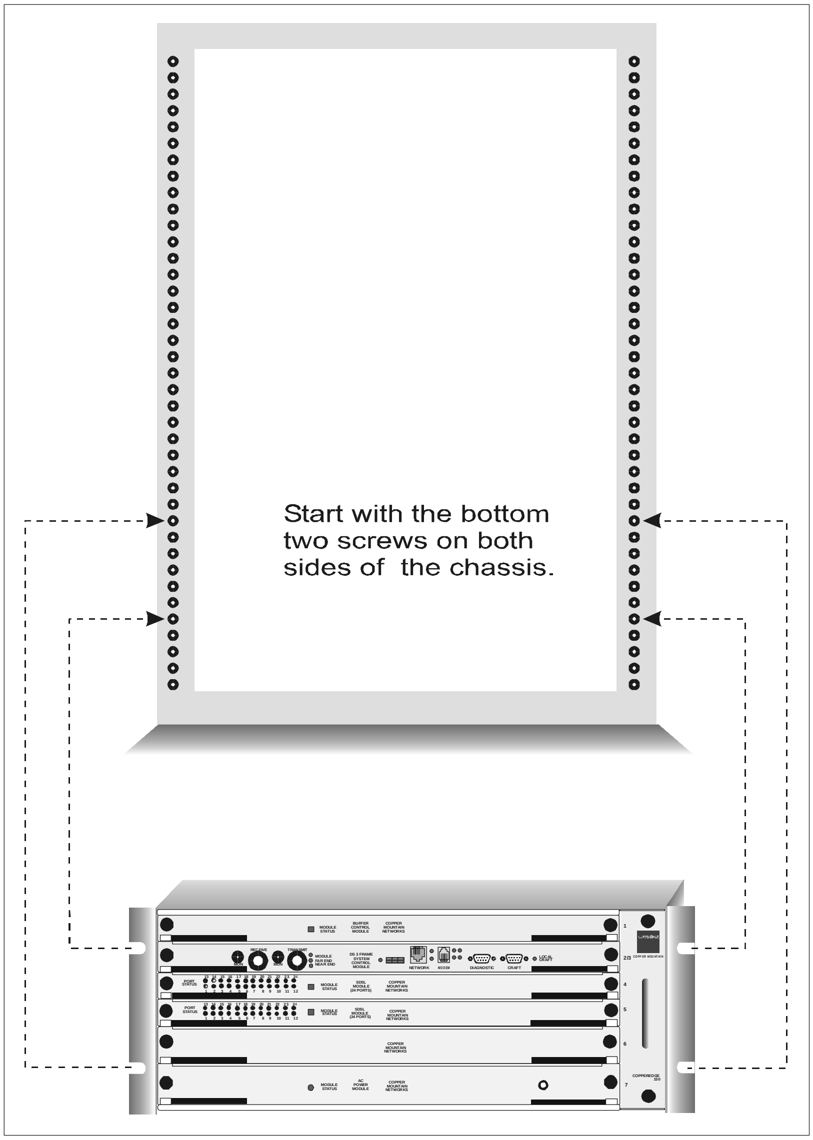
When you unpack your CE150 system, check that you have the following items:

- One of the following chassis/WAN combinations:
  – CE150 Chassis, Ethernet WAN
  – CE150 Chassis, Quad-T1 Frame WAN
  – CE150 Chassis, DS-3 ATM WAN
  – CE150 Chassis, DS-3 Frame WAN
- One or two DSL line modules installed in the chassis
- One AC or DC power module installed in the chassis
- Removable Fan Assembly
- Cables
- Optional mounting kit for rack installation
- User documentation

Rack Installation

The CE150 can be installed with attachment hardware for either 19-inch EIA or 23-inch Telco CO style racks. The standard configuration is 19-inch ears for mounting, although the CE150 has rubber feet on the bottom for sitting on a table.

A typical rack-mount installation is shown in the following diagram.



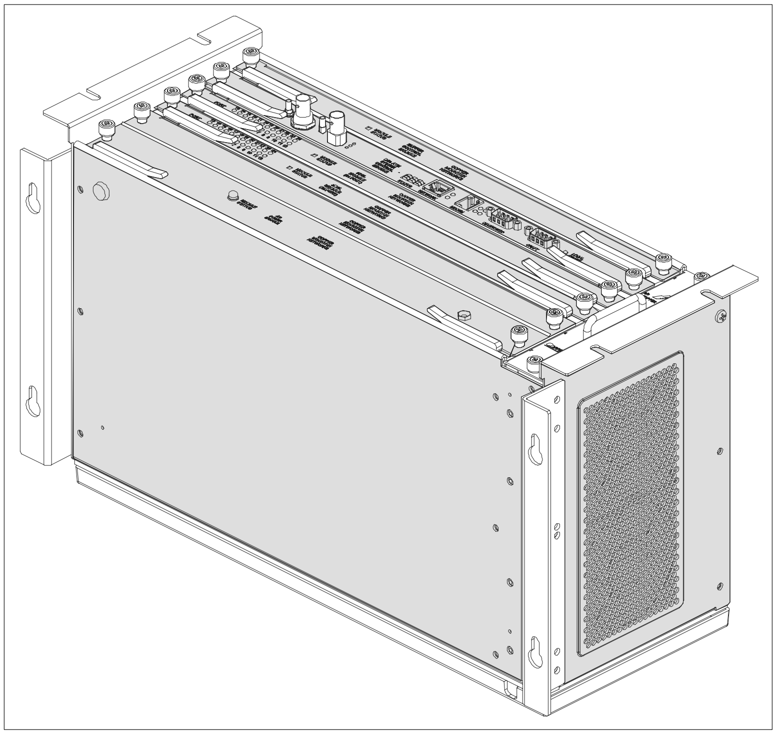
A Typical Rack Mount Installation

Wall Brackets

The CE150 can also be mounted flat against a wall, using two brackets attached to the wall. In this position, the LEDs and connectors for diagnostics ports, craft ports, and WAN ports are facing the ceiling. The connectors for DSL links are facing the floor.

Four screws (two for each of the brackets) attach the brackets to the wall. They are ¾-inch wood screws intended for installation into ¾-inch plywood. Six screws (three on each end of the CE150) attach the CE150 to the brackets.

A cable clamp also attaches to the back of the CE150. It supports the power cable, keeping it from being accidentally pulled out of the power connector.



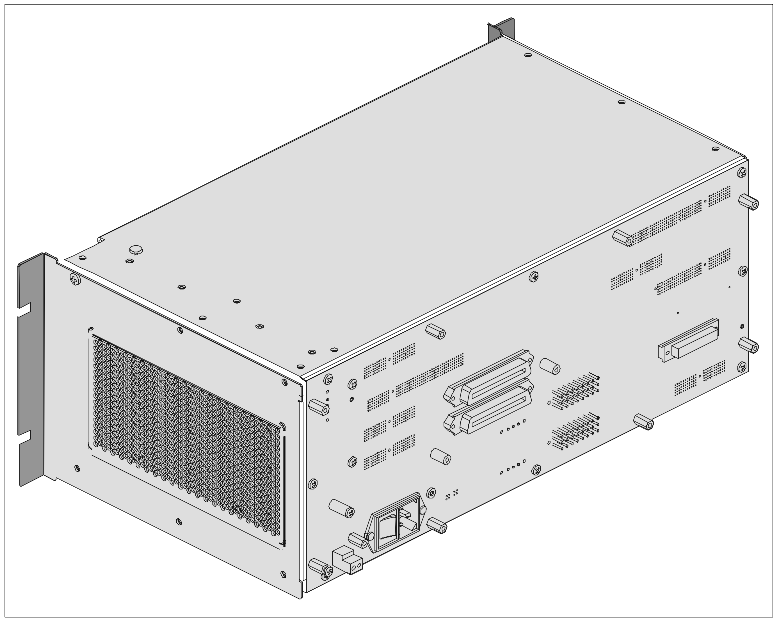
CE150 with Brackets for Wall Installation

For detailed instructions, see the Mounting Instructions in the tear-out template at the back of this manual.

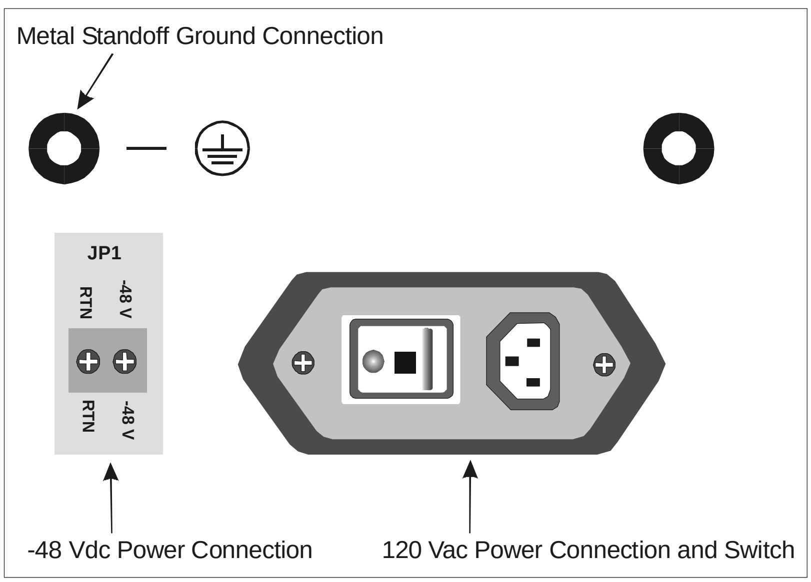
Rear Panel Connectors

The rear panel of the CE150 is a sheet of clear acrylic, separated from the circuit backplane by metal standoffs. In cutouts on the sheet of plastic are a series of connectors.

Also, just below and to the side of the 120 VAC power is a screw for attaching the cable clamp that supports the power cord or -48 VDC power cables.



Rear View of the CE150



Rear Panel Power and Ground Connections

N O T E

Either the -48 VDC or the 120 VAC power may be used, but not both

120 VAC Power Connection for North America

Connection of the CE150 to 120 VAC Mains is provided by an IEC 320 power connector with an On/Off switch, as shown in the previous illustration.

To connect 120 VAC power to the CE150, follow these steps:

1. Take the power cord that comes in the box with the CE150 and plug it into your power source.
2. Make sure the rocker switch beside the power plug is set to Off.
3. Plug the power cord into the 120 VAC power connector.

-48 VDC Power Connection

The CE150 may be connected to -48 VDC power systems in accordance with the following requirements.

Provide Over-Current Protection

The CE150 may be connected to either of the following -48 VDC power systems, however the maximum current must be limited to 20 amperes.

- If the CE150 is connected to an single -48 VDC power supply, the supply must comply with the regional Regulatory Safety Standards identified in the Certifications section of this manual, and internally current limit the -48 VDC to no more than 20 amperes.
- If the CE150 is connected to a “Bulk” -48 VDC power supply system, the supply system must comply with the regional Regulatory Safety Standards identified in the Certifications section of this manual, and external 20 Ampere Circuit Breakers must be installed between the Bulk -48 VDC power source and the CE150 -48 VDC power input.

Ground the CE150 Chassis

When operating on -48 VDC power, the CE150 chassis must be connected to a professionally installed Protective Earth, which is independent of the AC Mains Power Earth.

Specific instructions for installation of Protective Earth connections are contained in the National Electric Code for the United States and Canada. The Protective Earth must be installed by a licensed and qualified electrician.

Connection of the CE150 chassis to the grounding electrode must be made in conformance to the local and national regulations and may only be performed by trained and licensed service personnel. The grounding electrode conductor must be sized equivalent to a stranded 12 AWG wire.

DC Power Connection

Connection of the -48 VDC power system to the CE150 is provided by a barrel-type connector, as shown in the previous illustration.

To connect the -48 VDC power to the CE150, follow these steps:

1. Loosen the screws in the barrel-type terminals, named JP1, for both the -48 volt line and the return line.
2. Insert a red #12 AWG stranded copper wire for the -48 volt line into the hole in the bottom of the barrel marked -48V.
3. Connect the other end of the red #12 AWG stranded copper wire to the -48 VDC power source.
4. Insert a black #12 AWG stranded copper wire for the 48 volt return connection into the hole in the bottom of the barrel marked RTN.
5. Connect the other end of the black #12 AWG stranded copper wire to the 48 VDC power source Return connection.
6. Tighten both screws so the wires cannot slip out.

C A U T I O N

Reversing the polarity of the -48 VDC input connections will cause damage to the DC Power Module, requiring its return to the factory for repair. Never rely on wire color or "logic" when connecting the supply lines. Use an appropriate meter or testing device to verify the polarity before connecting the DC Power Sources.

Connection to Telecommunications Network Voltage Circuits

All CE150 Telecommunications Line Cards are compliant for connection to TNV-1 circuits, as defined in Section 6.2.1 & Annex V of UL 1950 3rd Edition and CSA C22.2 No. 950-95.

If CE150 Telecommunication Line Cards are used to communicate over TNV-2 or TNV-3 circuits, a protective device (such as a POTS Splitter), which provides isolation from TNV-2/TNV-3 voltages, is required.

For the United States and Canada, the Protective Device must be approved to UL 1950 3rd Edition and CSA C22.2 No. 950-95, NRTL Marked, and a Declaration provided stating that this protection is provided.

DSL Link Connections

The illustrations in this section show rear-chassis cable connectors and pinouts for wiring the DSL ports. Each connector (RJ-21X) services a single DSL module as indicated by the slot numbers. Use standard 25-pair cables of a length to connect the CE150 to your 66 blocks or other terminal point.

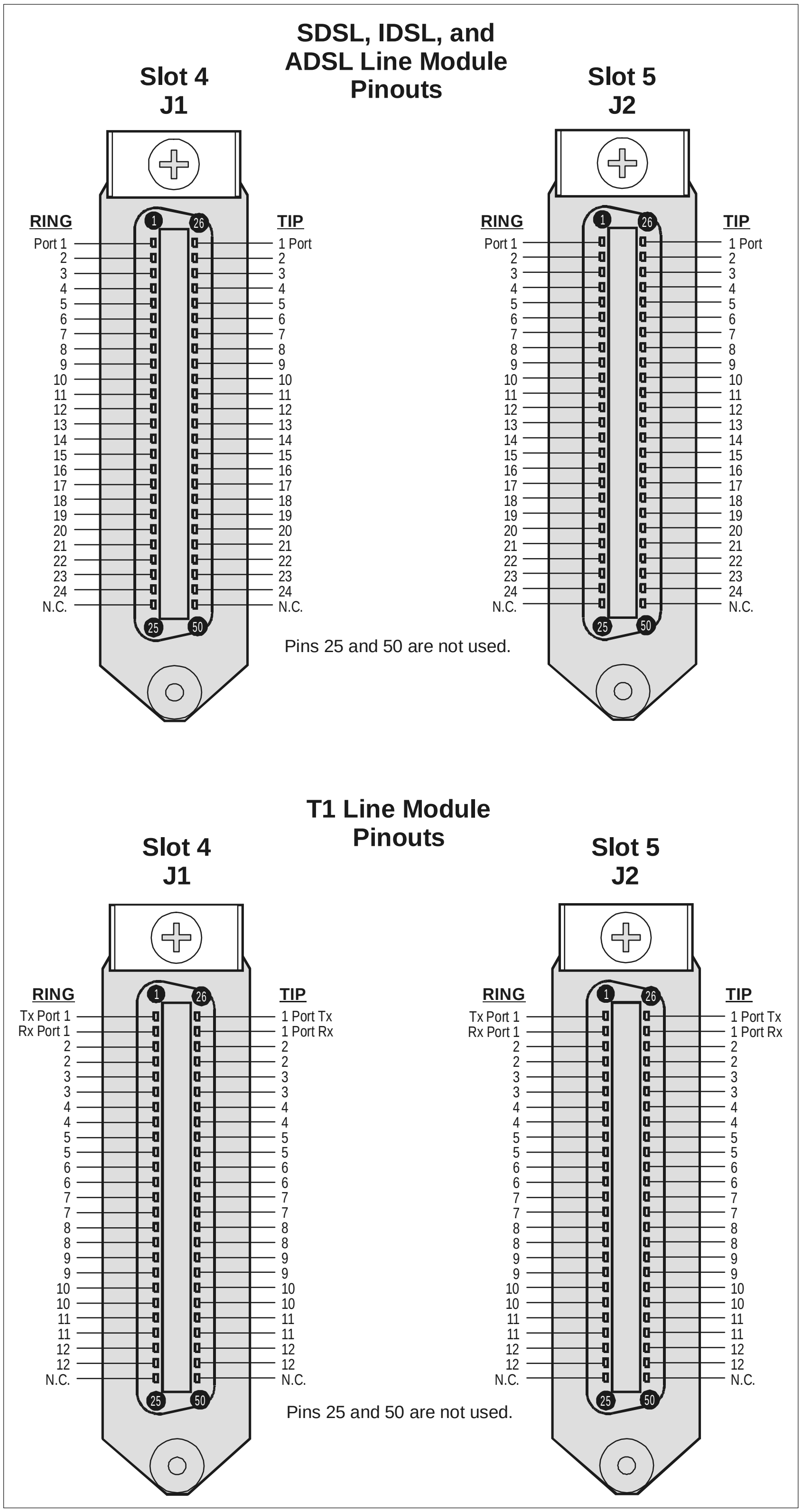
NOTE

The CE150 is designed for installation in facilities that provide over-voltage protection on the subscriber loops. This protection should be greater than or equivalent to that provided by three millimeter gap carbon blocks.

CAUTION

Use only waxed nylon string or nylon cable ties threaded through the white cable tie brackets to secure the attached amphenol connectors in place (in addition to the screw on the connector). If you use other types of attachment hardware, take care that they do not contact the CE150 backplane circuitry, and that cables are held securely in place. Damage resulting from use of non-standard hardware may not be covered by the warranty.

The CE150's Line module slots are filled at the factory in ascending slot-number order. The first DSL module is installed in Slot 4 (served by connector J1) and the second in Slot 5 (served by connector J2).



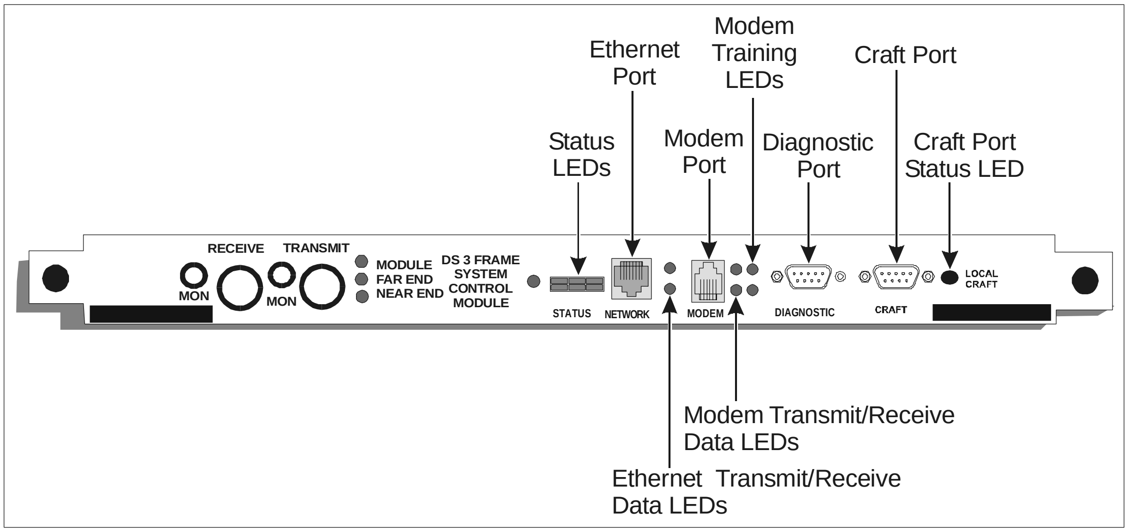
DSL Cable Connectors and Pinouts on the Back of the CE150

Front-Panel Connections

This section describes connections for the System Control Modules (SCM-1, SCM-2, and SCM-3) and WAN modules.

System Control Module 1 and System Control Module 2

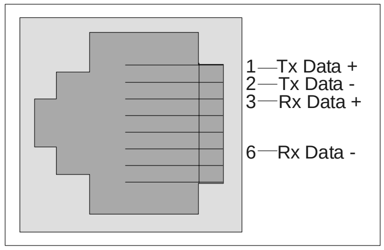
The System Control Module 1 and System Control Module 2 have the Ethernet, Diagnostic, and Craft ports for the CE150.



System Control Module

Ethernet Port

Connect a Category 5 STP cable (with RJ-45 connectors at both ends) between the Ethernet Port connector, and any available port (10BASE-T or 100BASE-T) of a network facility, such as a hub.

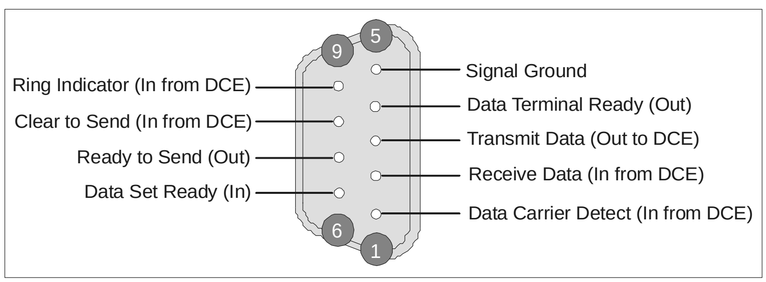


Ethernet Port Pinouts

Craft and Diagnostic Serial Ports

The Craft and Modem ports share the same circuitry and cannot be operated simultaneously. The Craft port takes precedence: if the two are in operation at the same time, the Craft port will function and the modem port will not.

Both the Craft and Diagnostic connectors are DB9, male, EIA/TIA-232 serial interfaces.



Pinouts for the Craft and Diagnostic Ports (EIA/TIA-232, DB9)

Connect a null modem cable between the COM port on the local terminal or computer and the Craft port. Note that the Craft port is a DTE interface. If you are connecting a terminal or computer which are also DTEs, you will need to use a null modem cable.

The Diagnostic port is intended to facilitate factory processes. If there is a problem with your system, however, your Copper Mountain Technical Support engineer may ask you to connect the Diagnostic port to the serial port of a local computer to aid in troubleshooting.

N O T E

We recommend that you do not connect the Diagnostic port except as directed by Technical Support. Restarting a computer with one of its serial ports connected to the Diagnostic port may cause the CE150 to reset.

Analog Modem Port

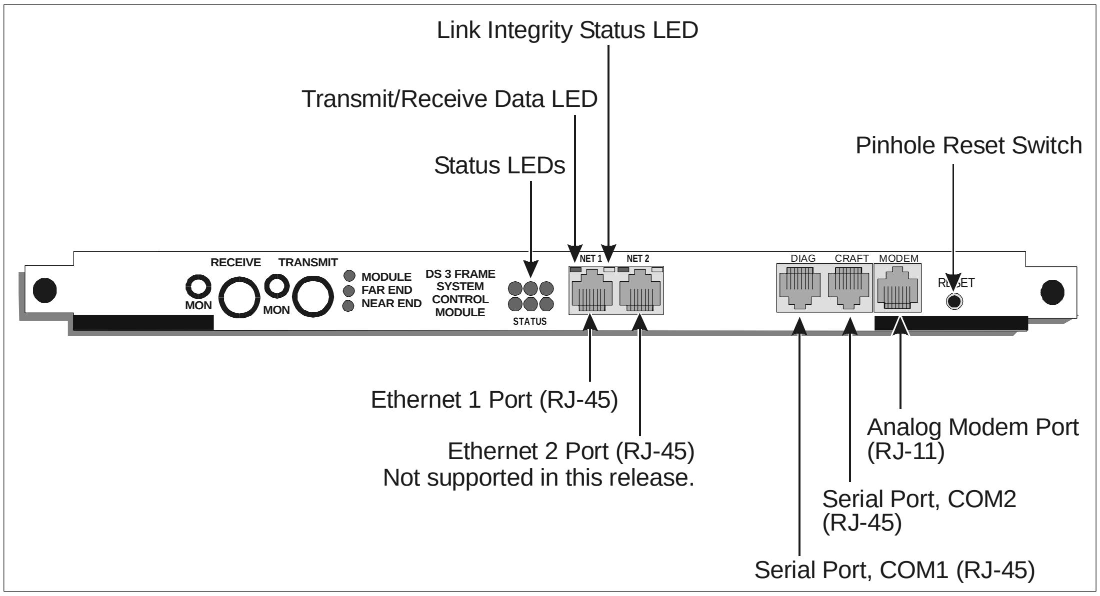
Connect the telephone cord (with RJ-11 connectors on each end) between the telephone outlet and the modem port on the CE150. Note the four LEDs adjacent to the modem port. The pair on top are green when the modem is training. The pair on the bottom are green when the modem is receiving and transmitting.

The Craft and Modem ports share the same circuitry and cannot be operated simultaneously. The Craft port takes precedence: if the two are in operation at the same time, the Craft port will function and the Modem port will not.

The modem is set to automatically answer on the second ring. However, it will take longer to answer when it is trying to reconnect after a disconnection—it cannot answer on the second ring because it is still busy disconnecting from the previous session.

System Control Module 3

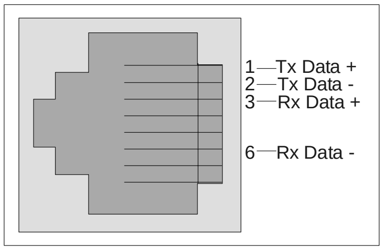
The System Control Module 3 differs from the System Control Module 1 and System Control Module 2 in that it has a second Ethernet port. Also, the Diagnostic and Craft ports are RJ-45, not DB9, connectors.



System Control Module 3

Ethernet Port

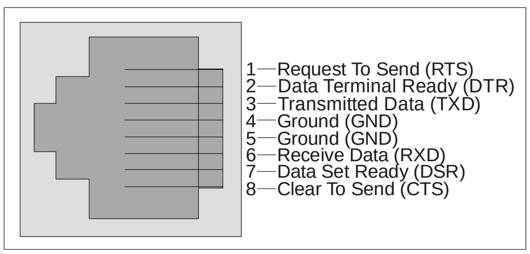
Connect a Category 5 STP cable (with RJ-45 connectors at both ends) between the Net 1 connector, and any available port 10BASE-T or 100BASE-T) of a network facility, such as a hub. The Net 2 connector for Ethernet 2 is not supported in this release.



Ethernet Port Pinouts

Craft and Diagnostic Serial Ports

Both the Craft and Diagnostic ports are 16C550 PC-Compatible serial ports with RJ-45 connectors. Unlike the SCM-1 and SCM-2 boards, you can concurrently use the Craft and Modem ports on the SCM-3 board.



Straight-through Pinouts for the Craft and Diagnostic Ports

If you are using an access server, connect a Category 5 STP roll-over cable (with RJ-45 connectors at both ends) between the COM port on the server and the Craft port on the CE150.

If you are using a local terminal or computer, attach a Category 5 STP cable (with RJ-45 connectors at both ends) between the COM port on the local terminal or computer and the Craft port on the CE150. Refer to the following table for the pinouts.

Cable Pinouts for the Craft and Diagnostic Ports

DB9 Connector	Signal	RJ-45 Connector
1	Data Carrier Detect (in from DCE)	Not used
2	Receive Data (RXD)	6
3	Transmit Data (TXD)	3
4	Data Terminal Ready (DTR)	2
5	Ground	4 and 5
6	Data Set Ready (DSR)	7
7	Ready (or Request) to Send (RTS)	1
8	Clear to Send (CTS)	8
9	Ring Indicator	Not used

The Diagnostic port is intended to facilitate factory processes. If there is a problem with your system, however, your Copper Mountain Technical Support engineer may ask you to connect the Diagnostic port to the serial port of a local computer to aid in troubleshooting.

N O T E

We recommend that you do not connect the Diagnostic port except as directed by Technical Support. If you restart a computer with one of its serial ports connected to the Diagnostic port, the CE may reset.

Analog Modem Port

Using a standard telephone cord (RJ11 connectors), connect the modem port on the CE150 System Control Module to a dial-up phone line. The port uses 8 data bits, 1 stop bit, and no parity bit.

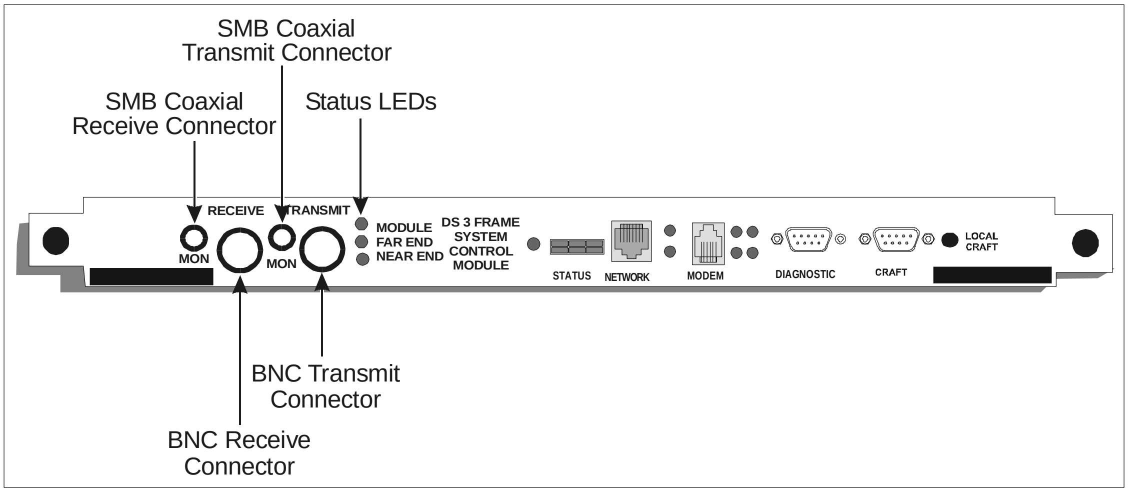
The modem is set to automatically answer on the second ring. However, it will take longer to answer when it is trying to reconnect after a disconnection—it cannot answer on the second ring because it is still busy disconnecting from the previous session.

If the modem connection is dropped and the operator did not log out of the Craft session, the Craft session is terminated within 15 minutes. The session is immediately available, without requiring a password, to the next person who dials in during the 15 minute period. *This is a security risk*; make sure all operators are aware that they need to properly log out of the Craft session.

Unlike the SCM-1 and SCM-2 boards, you can concurrently use the Craft and Modem ports on the SCM-3 board.

DS3 WAN Module (ATM or Frame Relay)

The DS3 WAN module (either Frame or ATM) has separate BNC connectors for the Transmit and Receive signals. Two RG59/U coaxial cables are provided to complete the connection between the module and the DS3 access device. Two subminiature coaxial connectors (SMB) allow test equipment to be connected to the Transmit and Receive Monitor ports.



DS3 WAN Module

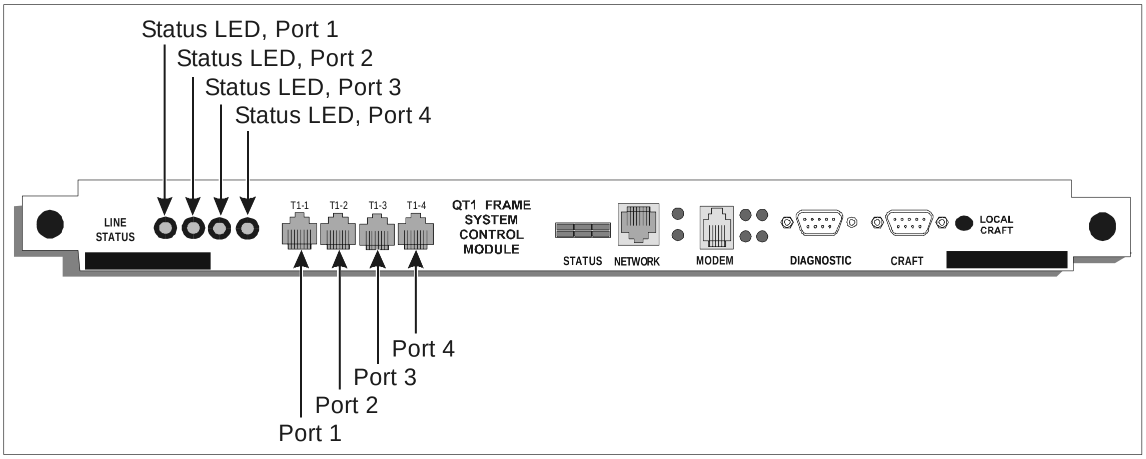
Three LED indicators are located above the Transmit BNC connector. At power-up, all three LEDs will illuminate briefly as a functional test. Each LED has a specific color to indicate its state:

DS3 WAN Module LEDs

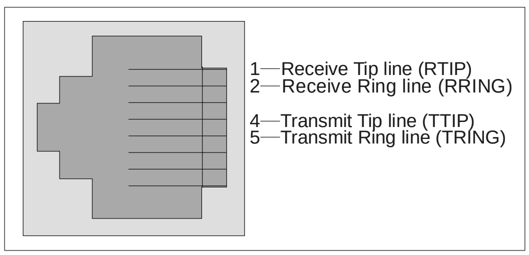
Indicator	Color	Description
Module	Green	Normal operation. The line has been synchronized with the network and shows no local or far-end alarms.
Far End	Yellow	A far end failure condition. The LED stays yellow until the condition is cleared.
Near End	Red	A near end failure condition, including either loss of signal or loss of frames. The LED stays red until the condition is cleared.

DS1 WAN Module (Quad T1)

The DS1 WAN module has four Quad T1 ports. Four unshielded twisted-pair cables (rated at 100 ohms) with RJ-48C modular jacks are provided to complete the connection between the module and the DS3 access device.



DS1 WAN Module



Quad T1 Pinouts

Each T1 line has an LED that indicates the condition of the line or the type of traffic on the line, including alarm signals and normal operating signals. The LEDs have three colors indicating different states.

DS1 WAN Module (Quad T1) LEDs

Color	Description
Green	Normal operation. The line has been synchronized with the network and shows no local or far-end alarms.
Yellow	A far-end failure condition. The LED stays yellow until the condition is cleared.
Red	A near-end failure condition, including either loss of signal or loss of frames. The LED stays red until the condition is cleared.

Pre-Power Check

Before you apply power to the CE150 for the first time, take a few minutes to check the physical integrity of the installed unit:

- Check that the power wires match the right leads. If you are uncertain, use the voltmeter to test the circuitry.
- Check that all plug-in modules are properly aligned and seated in their connectors.
- Check all modules to ensure that their extraction levers are lying flat, parallel to the plane of the front panel (module inserted position).
- Check the retainer screws on all plug-in modules and assemblies, including the fan tray and the alarm module. Use your fingers to tighten any loose screws until they are snug.
- Check all connectors to be sure they are correctly installed and fastened securely.

This completes the physical installation.

Power-Up LED Sequence

Before you connect power to the CE150, read this section so you will understand the LED start-up sequence.

Power-Up LED Sequence

After you turn on the CE150, the LEDs follow this start-up sequence:

1. On the AC Power Module (slot 7), the Module Status LED is lit with a green color.
2. As power is applied, Module Status LEDs on the Buffer Control Module and on the SDSL Modules cycle as follows:

   Red for ½ second
   Orange for ½ second
   Green for ½ second.

3. **Normal Start-up:**

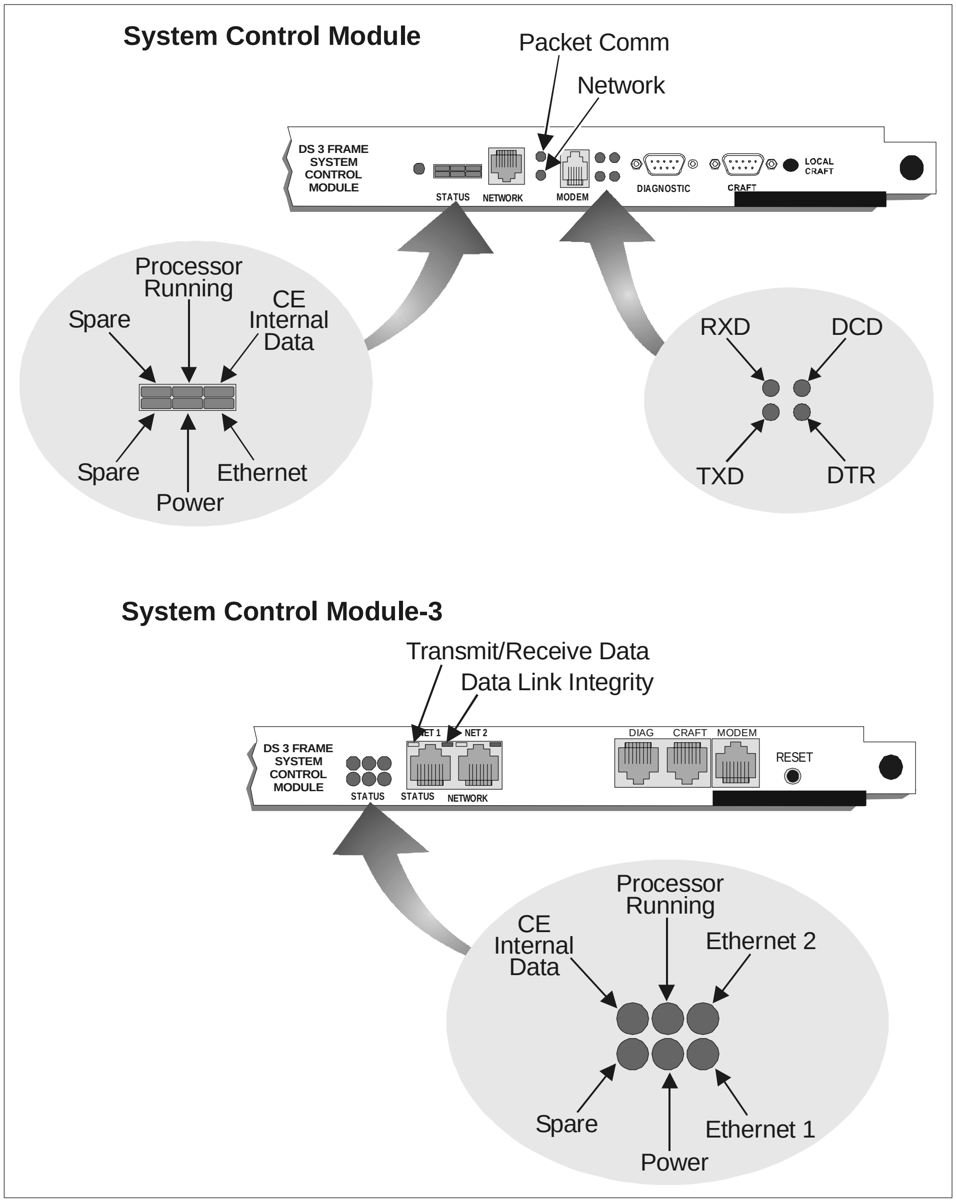
   After the cycle described in Step 2, the Buffer Control Module Status LED turns red, and then begins to blink. As the initialization process ends, usually taking less than a minute, the status LED changes to a rapidly flickering green. On the SDSL Modules, the Module Status LEDs turn amber while software is being downloaded to them from the System Control Module, then switch to steady green.

   If software is being installed:

   After the cycle described in Step 2, the status LEDs turn a steady amber until the software download begins. While the software download is in progress, the Buffer Control and DSL Module status LEDs turn to a blinking amber.

   When download is complete, the SDSL Module status LEDs turn green; the Buffer Control Module status LED changes from amber to rapidly flickering green.

4. The System Control Module and System Control Module-3 have six LEDs beside the Ethernet connector. When power is applied, the Power LED immediately turns green, and the Processor Running indicator turns green within a second or two.



System Control Module LEDs

5. If the System Control Module is connected to an Ethernet network, the module will synchronize and negotiate its speed with the network. The Ethernet LED will turn green if the network is a 10BASE-T (10 Mbps) system and red if it is a 100BASE-T (100 Mbps) system.

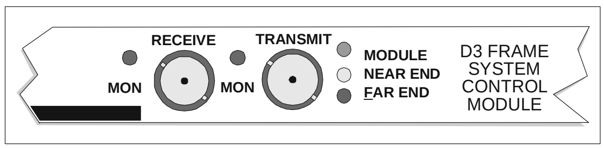
   The CE Internal Data LED indicates data activity internal to the CE150 itself. The state of this LED does not correspond to the status of the system. At any time, the Internal Data LED may be off, blinking, pulsing, or solidly green

   On the System Control Module panel, the Network and Packet Comm LEDs beside the modem port indicate

data traffic. The Network indicator turns or flickers green when Ethernet data packets are transmitted by the CE200. The Packet Comm indicator turns or flickers green when Ethernet data packets are received by the CE200.

On the System Control Module-3 panel, the Ethernet port has two LEDs: a single Transmit/Receive Data LED and a Data Link Integrity LED. The Transmit/Receive Data LED indicates data traffic. It turns or flickers yellow when data is received or transmitted on the Ethernet line. The Data Link Integrity LED turns green when the integrity of the Ethernet link for 10BASE-T or 100BASE-TX is acceptable.

6. The DS3 WAN Module (if present) has three LED indicators on its front panel, just below the Transmit BNC connector. At power-up, all three LEDs will light briefly as a functional test.



DS3 Frame Module LEDs

If the DS3 (ATM or Frame) WAN module is connected to a compatible DS3 facility, then the green (Module) LED will light and the Red (Near End) and Yellow (Far End) LEDs will be off.

In the event of a local failure such as loss of signal or frame errors at the CE150, or within the DS3 Module itself, the Near End LED will light red to indicate the failure. If a failure occurs that is not associated with the CE150 or with the DS3 Module, but instead is the result of a failure on the link or with upstream equipment, then the Far End LED will light yellow to indicate that failure mode.

7. If there is a saved configuration file, it will be read in automatically. If there are CopperRocket CPE units connected to the CE150's SDSL ports, they will attempt to train (handshake and connect) on the CE150, *regardless of whether the ports have been configured*.

As the Customer Premises Equipment attempts to train on a port, the Port Status LED on the corresponding SDSL Module will first blink rapidly, and then turn steady green if the CPE trains successfully.

If the CPE attempts to train but fails, the Port Status LED will go off until the CPE again tries to train. *[It may take several minutes for all of the connected ports to come up.]*

The Port Status LEDs of unconnected DSL ports will periodically blink green as the system checks their status.

If you encounter problems while starting the system normally (such as from data stored in the CE150's flash memory), please contact Copper Mountain Technical Support.

Connect the Power

Now that you understand the LED start-up sequence, apply power to the CE150 and observe the LEDs.

Check the Fan Tray

After the CE150 has successfully been powered on, check the fan tray to be sure it is operating properly.

Hold your hand beside the chassis and feel for blowing air.

If you cannot feel any blowing air:

1. Disconnect the power from the CE150.
2. Use the voltmeter to recheck the wiring.
3. Reapply power to the CE150.

If the power is good but you still cannot fell any blowing air, disconnect the power to the CE150 and contact Copper Mountain Technical Support.

Installing or Replacing Modules

For information about replacing modules, see Chapter 8.

Chapter 3
Initial Configuration

This chapter describes procedures for performing initial provisioning and other basic configuration tasks. Before you attempt to modify any of the software settings, however, you should be familiar with the sections on General Configuration Guidelines, Command Structure, and the Permanent Interface Identifier scheme of specifying interfaces on the CE150.

The term *DSL module* is generally used throughout this chapter to mean *any supported* DSL module: SDSL ADSL (G.lite or G.dmt), or T1. If there is a difference in the way one or the other types of DSL modules behave or what they require, it will be explicitly specified.

System Configuration Guidelines

Configuration of the CE150 is accomplished by entering SNMP-like commands to modify the Management Information Base (MIB). This section contains instructions for querying and configuring the CE150 as part of the installation process.

Accessing the CE150

You can log in to the CE150 using the Command Line Interface (CLI) or the Graphical User Interface (GUI):

- You can use CopperView or some other SNMP-based NMS to remotely configure the CE150.
- You can use a LAN and a telnet session to access the CLI to remotely configure the CE150.
- You can use a terminal (or terminal emulator) to the Craft serial port on the SCM front panel and then employ the CLI to configure the CE150 on site.

To access the CopperCraft command line interface (the CLI) through your PC COM port, use Hyperterminal or an equivalent terminal emulation application. Set the serial port rate to 9600 bps, 8-N-1 (8 data bits, no parity, 1 stop bit). If you want to use flow control, choose the hardware (RTS/CTS) settings.

CE150 installation, on-site verification, and establishment of upstream data links are normally done using the CopperCraft interface, either through direct connection method or through Telnet. Thus, this document puts greatest emphasis on those methods.

For information on the CopperView Element Manager and its graphic user interface, see the *CopperView Element Manager, Installation and Operation Guide*.

CopperCraft Login

As the CE150 boots, the Copper Mountain Networks copyright and system information is displayed, followed by the user name prompt.

```
Copper Mountain Networks
CopperEdge200
Copyright 1996 - 2001, All Rights Reserved
System Name: Tech Pubs
Build Date: Feb 27 2001, 20:29:54

USERNAME>
```

To log in, enter **admin** for both user name and password.

```
USERNAME> admin
PASSWORD> *****
```

Once you are logged in, the factory default command line prompt is displayed:

```
ce150(1.3)>
```

This default prompt consists of the system name (ce150), as defined in the Name object of the cmSystemTable, plus the slot number of the active System Control Module (1.3). Since all CE150s initially have this default prompt, we recommend you change the system name in the cmSystemTable to a more meaningful name. You cannot change the slot number.

In this manual, we show the command line prompt as:

```
CRAFT>
```

CopperCraft Logout

When you are ready to log out of the system, enter **`exit`**, **`quit`**, or **`logout`**.

If you changed any system settings and did not save the configuration, the following prompt is displayed:

```
CE configuration has been changed.
Save the configuration?(Y / y)>
```

Enter `Y` or **`y`** to save the configuration. If the save was successful, the following messages are displayed:

```
.....
CE configuration save succeeded.

2000/11/14-15:33:17  USER LOGGED OUT
```

If you enter a response other than Y/y, or if you do not respond within 30 seconds, the configuration will not be saved, but the logout process will continue.

If you entered Y/y to save the configuration but do not want to wait for the save process, press **Esc**. The configuration will be saved and the logout process will continue.

Command Line Interface

In general, commands and other data can be entered without regard to case; either capital letters or lowercase letters are acceptable. However, to specify a directory path on a file server, use lowercase letters, as many servers are case-sensitive and will not download their software if commands are not lowercase.

If a long command line wraps, or for any reason you are in doubt about what the entire line actually contains, and in what order, type ^L (Control-L) to refresh the display of the entire command. For more information about line editing, see *CopperCraft Line Editor* on the next page.

CopperCraft Line Editor

Command strings for SNMP object groups can be very short or very long, especially when you are setting a series of new parameters in an object group. Here is an example of a short getall command, which returns data about all of the boards in the CE150's chassis:

```
CRAFT> getall cmboard
```

Here is an example of a longer Set command, which sets the parameters on an ATM quality of service table:

```
CRAFT> set cmcircuitparam [6] rowstatus=createandgo
servicecategory=nrtvbr pcr=104000 scr=604 mbs=4
cdv=unspecified
```

The line editor function on the CopperCraft CLI lets you recall as many as 20 previous commands from the system's memory buffer. You can move through the command lines and edit them before you press Enter and send the command to the system.

To display the list of valid Line Editor commands, press **esc-?** at the system prompt.

```
CRAFT>
^D           Delete current character
DEL          Delete current character
^H           Backspace
^E           End of line
^A           Beginning of line
Escape F     Forward one word
Escape B     Backward one word
^K           Delete to end of line
^U           Delete to beginning of line
^U^K         Delete entire line
^L           Redisplay current line
Left Arrow   Cursor left
Right Arrow  Cursor right
Up Arrow     Scroll up through history
^P           Scroll up through history
Down Arrow   Scroll down through history
^N           Scroll down through history
```

Control characters are shown with a caret (^). Press and hold the Control key, press the letter key, then release both keys.

Escape sequences are indicated by the word, Escape. Press and release the Escape key, and then press the letter key.

If you are using a terminal emulator, be sure to set it to VT100 mode. If a command line will not fit on a single line, an arrow character (< or >) appears at either end of the line indicating Scroll mode. Use the arrow keys to move to the end of the line; the next 10 characters will be displayed.

Permanent Interface Identifier (PII)

Fully populated CE150s can have over 48 physical interfaces, including DSL ports, WAN ports, and Ethernet ports. They can have an even larger number of virtual circuits. All ports and circuits must be uniquely identifiable so they can be configured, and so the traffic over them can be monitored. To accomplish the job, the CopperCraft CLI uses a system called Permanent Interface Identifiers (PIIs).

The PII system consists of four elements, which must be written in a series format in a specific order: c.s.p.v, where the letters represent the following:

c = Chassis shelf number (always 1)
s = Slot number (1 to 7)
p = Port number (For DSL, 1 to 24; for WAN, 1 to 4)
v = Virtual circuit number (1 to 976)

Slots in theCE150 chassis are reserved for specific types of modules, as shown in the following table:

Slots in the CE150 Chassis

Slot No.	Module
1	Buffer Control Module
2	DS1 or DS3 WAN Modules (Frame, ATM)
3	System Control Module (Ethernet Port)
4	SDSL, ADSL, or T1 Module (24 Ports)
5	SDSL, ADSL, or T1 Module (24 Ports)
6	Currently unused
7	Power Module

If the ports in slot 2 are on a their numbers are 1 to 2 from top to bottom. If the ports are on Quad T1 boards, their numbers are 1 to 4 from top to bottom. Slot 3 always should have a System Control Module in it; slot 1 always should have a Buffer Control Module in it. The CE150 system, will not function without a System Control Module in slot 2 and a Buffer Module in slot 31.

The PII system of identifying the interfaces on a CE150 works in this way:

- A PII such as 1.3.1 identifies the CE150 in chassis 1, the System Control module in slot 3, and port 1 (the Ethernet port).
- A PII such as 1.2.1.24 identifies the CE150 chassis 1, the WAN module in slot 2, port 1, and virtual circuit 24 (it could be either Frame Relay, ATM, or Quad T1).

- A PII such as 1.4.24 identifies the CE150 chassis 1, the DSL module in slot 4, and port 24 (it could be SDSL ADSL, or T1).

If you are using an SNMP manager to configure the CE150, you can use a slightly different PII format. In such a case, you probably will enter the identifier without periods to separate the segments, and you probably will fill parts of the fields with leading zeros. For example, a PII of 1.2.1.24 entered at the CopperCraft CLI will become a PII of 102010024 entered in an SNMP manager.

SNMP Command Structure

Command strings for SNMP object groups must have certain elements in them, which must appear in a certain order. If elements are missing or if they are out of order, the CE150 will not be able to process the commands that you sent through the CopperCraft CLI. Instead, the system will issue an error message.

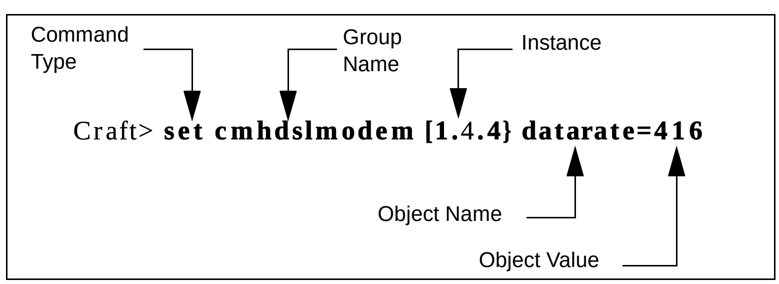
Command strings for SNMP object groups have a minimum of two elements:

- a command type
- an object group name

Command strings for SNMP object groups have a maximum of five separate elements:

- a command type
- an object group name
- an instance (typically a PII)
- one or more object names
- an object value for each object name

Here is the structure of a typical Set command that is assigning a data rate on an HDSL port of 416 Mbps. The port is the fourth one on a board in the fourth slot of the chassis:



Example of a Set Command

You must enter the elements of command strings in the following order, and you must adhere to the following restrictions:

- Command type: *set, get, getnext, getall, help, find, ping*.

  Always enter the command type first, separated from the group name by a space.

  For operators with Security or System privileges, the following commands are also available: elog, alarms, LCRestart, and SCMRestart.

- Group name: *cmSystem, cmIface, cmADSLModem*, etc.

  Always enter a group name for the set, get, getnext, getall, and find commands. Otherwise, the system will return an error message. All of the group names are listed alphabetically in the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual.

- Instance: *IP address, PII*, etc.

  Enter an instance for a Set or Get command. A port—either a WAN port or a DSL port—can be divided into VCs. To identify the VC, include the number of the VC after the number for the slot. For example, the instance, 1.2.1.30, refers to the WAN module in slot 3 and the first port. The number, 30, identifies the VC.

  Always enclose an instance in square brackets [ ]. If an entry requires more than one index, place the indexes in the same order as displayed in the help objects listing. Always separate multiple indexes with a comma, but always enclose them with a set of brackets: [*index1, index2*]. Multiple indexes form a single instance.

- Object name: *NetModel, IpAddr, NetMask*, etc.

  Enter an object name when performing a Set command. (In the three different Get commands, object names are unnecessary.) Separate the object name from the instance by a space. Also, an object name must be accompanied by an object value.

- Object value: *possible values or value ranges are specified in each of the object group tables*.

  All of the objects and their values or value ranges are listed in the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual. In the command string, separate an object value from an object name with an equal sign (=), in this way: **netmodel=coppervpn**.

  Do *not* use spaces between object names and object values.

  In a Set command, you will be assigning values to multiple objects. In these situations, separate different objects (and their values) from each other with spaces, as follows:

  ```
  CRAFT> set cmiface [1.4.24] netmodel=vwan
  destpii=1.2.1.500 encapsulationtype=rfc1483
  cmcpcompatible=yes
  ```

You can enter multiple Get commands on the same line by separating them with a semicolon, as in the following example:

```
CRAFT> get cmiface [1.4.24]; get iftable [1.2.1.109]
```

Any of the CopperCraft object names can be truncated up to the point of its shortest unique character string. For example, in the sample Set command above, cmIface could be abbreviated to cmif, and ifTable could be abbreviated to ifta.

Some object names may also be entered using a shorter alias. Alias strings that differ from the object name, and that consist of something other than a truncated version of the name, are shown in the MIB-definition tables in the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual.

Whenever you assign IP addresses or subnet masks for the CE150, accuracy is essential; always issue a Get command to verify the entry when you are done. The CE150 will report an error if you try to assign the same IP address to more than one of its interfaces, but it does not currently report an error if two IP addresses for the same subnet are assigned to separate interfaces.

Helpful Shortcuts: Getall and Find

In addition to the standard SNMP commands, the CopperCraft interface provides two special command, Getall and Find. They are briefly described below. For more complete information about these commands, see the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual.

Getall

The Getall command allows you to review the status of all objects for all instances of a specified group at the same time. The output is more condensed than the output for a Get command.

The basic syntax is:

```
CRAFT> getall mib_group
```

Since the list could have hundreds of entries, the screen will fill up and stop at four or five entries. To view additional entries, press any key except Esc. Continue paging until you find the entries you want or until you reach the end of the list. To quit the listing at any time, press Esc.

Find

The Find command provides an easy way of compiling lists of interfaces based on common criteria. The lists are useful for troubleshooting and record-keeping. You can use the find command in several different ways.

To specify up to three objects and their values for all instances, the basic syntax is:

```
CRAFT> find mib_group object1=value1...object3=value3
```

To specify up to three objects and their values starting with a specific instance, the basic syntax is:

```
CRAFT> find mib_group [instance] object1=value1...
object3=value3
```

You can include up to five additional object names after the three objects and values to refine the output. The system will return only the values for the instance and the object names for those entries that matched the specified criteria.

Initial Configuration

With the CE150 successfully started, you can now configure the CE150, associated host devices, and links (both DSL links and upstream WAN links). Initial configuration consists of setting up attributes for two-way communication over the network.

The required steps, described in the following sections, consist of:

1. If your CopperEdge Concentrator is equipped with a DS3 Protection Switch/Alarm Panel, test the redundancy performance.
2. Setting up records (user names and passwords) for authorized system operators
3. Storing the CE150 system name, location and contact information (optional)
4. Setting the system clock
5. Specifying an SNMP manager to which traps will be sent (a trap destination)
6. Configuring the CE150 network interfaces to enable remote access for system management

The *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual contains a reference of all of the CE150 MIB groups and their objects, and additional information about configuring or viewing them.

Configure Operator Names

This section describes the method of configuring and managing operator names and passwords through a local database internal to the CE150: the cmOperator MIB group. For more information about the cmOperator group, see the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual.

The initial tasks to be performed include:

- Create at least one unique operator name and password.
- Change the password for the factory default operator name (admin).
- If your system will be managed locally (not through an external Radius server) and an SNMP manager will be used, create the SNMP Community names.

Note the following restrictions for entering operator usernames and passwords:

- Do not use the following characters: backslash (\), double quotes (“ ”), or ending bracket (]). The CE150 may misinterpret them as other commands.
- The # character is not allowed.
- Spaces are not allowed in UserName strings. If first and last names are needed, use upper case or an underscore to separate them: JohnDoe or john_doe.
- Passwords are not completely case sensitive; the first character of a password may be accepted regardless of case. The system is case-sensitive to all successive password characters.

In addition to the username and password, you will set the operator’s context (authorizing system access through SNMP, nonSNMP, or All) and Privilege level (View, Monitor, or Provision). Only one operator, the Security Administrator, is allowed the highest privilege level (Security). The default operator (admin) is always the Security Administrator for the system.

You can use the Help command to view the list of available configuration options for each cmOperator object.

```
CRAFT> help cmoperator
```

Management by a Radius Server

If your system will be managed by an external Radius server, you will not create the SNMP Community names from CopperCraft. The user and password databases on all CE150s on the network, including CopperView (SNMP) sessions, will be consolidated on the Radius server. All login requests for CE150s will go to the Radius server for verification

Once the Radius server is provisioned and enabled, security features are under its control. All operators will be created on the Radius server instead of through CopperCraft; you can still configure the CopperCraft cmOperatorTable, but its contents will no longer be used. Operators configured in the cmOperatorTable and not contained in the Radius server database will not be able to log in. The Radius protocol requires that operator names and passwords are fully case-sensitive.

Create a Unique Operator

Create at least one unique operator name and password with the CLI using the Craft port, Telnet, or the modem; you cannot use SNMP or HTTP. We recommend you use this operator to perform the remaining configuration tasks.

To create a new operator:

1. From the control console, enter the name, password, state, context of nonSNMP, and privilege level of Provision.

   ```
   CRAFT> set cmoperator [mick] password=stones
   state=create context=nonsnmp privilege=provision
   Set Successful
   ```

2. Verify the new operator configuration:

   ```
   CRAFT> get cmoperator [mick]
   Group: cmOperator
   Instance: [Mick]
   Name                = Mick
   Password            = *******
   State               = active
   Context             = nonSNMP
   Privilege           = Provision
   ```

   Passwords are not displayed. If a user forgets a password, the Security Administrator can change it to another one, or else delete the operator and start again.

3. Test the new operator by logging out of the Craft session and logging in with the new username and password.

Change the Factory Default Operator Password

To ensure continued system security, you must change the password for the factory default operator name. Otherwise, unauthorized users could gain access to your system, with full privileges to change all settings.

Passwords are not completely case sensitive; the first character of a password may be accepted regardless of case. The system is case-sensitive to all successive password characters.

To change the password:

```
CRAFT> set cmoperator [admin] password=b5c312r
Set Successful
```

Verify the change:

```
CRAFT> get cmoperator [admin]
Group: cmOperator
Instance: [admin]
Name                  = admin
Password              = *******
State                 = active
Context               = All
Privilege             = Security
```

N O T E

If your system will be managed through an external Radius server, skip the next section. The SNMP Community names will be created on the Radius server.

Create the SNMP Community Names

To use CopperView AMS, CopperView EM, or another SNMP-based manager, you must first configure two operators (any community names, but typically, *public* and *private*). The first operator must have the View privilege, and the second must have the Provision privilege. They are to be used in the Read Community and Write Community login prompts, respectively.

To create the Public community name:

1. From the control console, enter the name, context, state, and operator privilege level.

```
CRAFT> set cmoperator [public] state=create
context=snmp privilege=view
Set Successful
```

2. Verify the configuration:

```
CRAFT> get cmoperator [public]
Group: cmOperator
Instance: [public]
Name                  = public
Password              = **********
State                 = Active
Context               = SNMP
Privilege             = View
```

To create the Private community name:

1. From the control console, enter the name, context, state, and operator privilege level.

```
CRAFT> set cmoperator [private] state=create
context=snmp privilege=provision
Set Successful
```

2. Verify the configuration:

```
CRAFT> get cmoperator [private]
Group: cmOperator
Instance: [private]
Name                 = private
Password             = **********
State                = Active
Context              = SNMP
Privilege            = Provision
```

System Information

Although not essential for operation, you should identify this CE150 by configuring the following objects in the System group.

```
Name = A unique identifier for this CE150
Location = Site information (CO, city, state)
Contact = Point of contact for this CE150 (name, title,
phone)
```

Example:

```
CRAFT> set system name=newname location=“e st. collo”
contact=“network administrator 800-555-4141”
Set Successful
```

1. Use quotes for character strings that include spaces.
2. Character strings cannot contain backslash (\), angle brackets (< and >), or apostrophe (‘) or pound sign (#) characters.

C A U T I O N

Using a pound sign (#) when configuring ports and VCs may cause the CE150 *to not read its config file correctly on reboot.*

3. The Craft command line prompt automatically changes to display the system name. It also displays the slot number of the active System Control Module. For example:

```
NEWNAME(1.3)>
```

N O T E

In this manual, examples of the Craft command line prompt do not show the slot number.

System Clock

Check the system clock and observe the date and time displayed in the format shown (time in 24-hour format) in the example:

```
CRAFT> get cmsystem calendartime
CalendarTime = 2000/06/15-12:04:17
```

If necessary, set the clock to the correct date and time:

```
CRAFT> set cmsystem calendartime=2000/06/15-15:05:00
Set Successful
```

Verify that the clock displays the correct date and time:

```
CRAFT> get cmsystem calendartime
CalendarTime = 2000/06/15-15:05:09
```

Trap Destination

Configure at least one trap destination (a managing device to which SNMP traps should be sent).

```
CRAFT> set cmtrapdest [n.n.n.n, 162] community=trapcomm
rowstatus=active
```

where **n.n.n.n** is the IP address of the SNMP manager.

Depending on the SNMP manager you are using, it may also be necessary to configure an IP port number or a community string for the receiver. For more information, see the description of the cmTrapDestination group in the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual.

Configuring the CE150 on the Network

To communicate over the network with a remote host or to set up a server to connect to the CE150 through Telnet or FTP, you must enable one of the system's network interfaces. It may be the Ethernet interface (1.2.1), one of the WAN interfaces in slots 3 or 4, or a DSL port.

1. Configure the network interface to be used for control and management of the system with an IP address and mask:

   ```
   CRAFT> set cmiface [1.3.1] netmodel=ip
   ipaddr=204.16.191.254 netmask=255.255.255.0
   encapsulationtype=none
   ```

2. If this interface will use a NetModel/ForwardingMode other than Full IP, specify those options as necessary.

3. Note also that if the network interface is a Frame Relay or ATM virtual circuit, you must also specify the DLCI or ATM Virtual Link number in the PII:

   ```
   CRAFT> set cmiface [1.2.1.23]
   farendaddr=202.114.66.188 netmodel=vwan
   encapsulationtype=rfc1490
   ```

4. In this example of an interface configured for VWAN, configure and activate the Frame Relay virtual circuit as described on page 65. Note that full provisioning of a Frame Relay or ATM WAN link may require additional configuration, depending on the type of digital facility used, connected upstream equipment, etc. See the discussions in Chapter 4 and the information on MIB groups related to WAN link provisioning in the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual.
5. Verify the physical connection of the CE150 network interface (Ethernet, WAN, or DSL), and ensure that any third-party equipment (router, CSU/DSU, etc.) is configured to communicate with the CE150.
6. Configure the CE150's default IP route. In most cases, the routes to connected networks that are followed by outgoing message packets are automatically defined by the IP addresses and subnet masks of the interfaces. For any messages that do not have an assigned routing, the default IP route (specified as 0.0.0.0) establishes the "next hop" address. This default route is mapped to its next hop through the port on the CE150 used for "upstream" packet-based communication. Typically this will be one of the WAN ports, but could be the Ethernet port, if that interface is used for network access. For example:

   ```
   CRAFT> set iproute[0.0.0.0] nexthop=192.176.1.88
   ```

   where: **192.176.1.88** is the IP address of the router interface to which packets for unknown destinations are routed. The mask of the default route is always 0.0.0.0. If no default route is configured, packets addressed to unknown destinations are discarded.
7. Configure other entries in the IpRoute table as necessary.
8. For distant destinations, you may need to insert additional routing-table entries. In fact, any other IP Route information, beyond the default, must be configured as static routes, as there is no provision for dynamic route assignment in the CE150. If additional routes are needed, add them to the ipRoute table as described in the previous section.
9. Using the CopperCraft interface, first view the existing route table, beginning with the default (0.0.0.0) route:

   ```
   CRAFT> get iproute [0.0.0.0]
   Group: ipRoute
   Instance: [0.0.0.0]
   Dest          = 0.0.0.0
   IfIndex       = 2
   Metric1       = 0
   Metric2       = -1
   Metric3       = -1
   Metric4       = -1
   NextHop       = 192.176.1.88
   Type          = Direct
   ```

```
Protot        = netmgmt
Age           = 64
Mask          = 0.0.0.0
Metric5       = -1
Info          = 0.0
```

The ipRouteTable is indexed by the destination IP address. A route entry requires at least a destination address and a Next Hop address. The Next Hop address specifies where to send a packet with the specified destination address.

For more details on ipRoute group and its objects, including valid responses and ranges for each, see the reference listing for the ipRoute group in the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual.

10. Save the interface configuration to the CE150's flash memory:

    ```
    CRAFT> set cmsystem command=save
    ```

11. Verify that the configuration information has been saved:

    ```
    CRAFT> get cmsystem  configsynch
    Group: cmSystem
    ConfigSynch          =  Saved
    ```

12. The CE150 responds with the cmSystem configuration. The CommandStatus object should display `Succeeded`. If Command Status shows `InProgress`, repeat the **`get cmsystem`** command; if Command Status shows `Failed`, contact Technical Application Support.

Chapter 4
Advanced Configuration

This chapter describes procedures for establishing DSL and network interfaces, and performing other configuration tasks. Important tasks include:

- Establishing subscriber DSL links (that is, configuring the Interface Table entries for each of the CE150's connected DSL ports).
- Configuring the upstream (WAN side) links, and setting up any additional parameters as needed for your specific operation.

The term *DSL module* is generally used throughout this chapter to mean *any supported* DSL module: SDSL ADSL (G.lite or G.dmt), or T1. If there is a difference in the way one or the other types of DSL modules behave or what they require, it will be explicitly specified.

Configuring DSL Links for Network Models

Configuration of DSL links (SDSL, ADSL, and T1 line modules) on a CE150 focuses on networking models (netmodels) and encapsulations. You must choose a netmodel for each DSL (and WAN) link. You can configure a CE150 with the following netmodels, depending on what your applications are:

- IP Routing
- Virtual Wide Area Network (VWAN)
- CopperVPN
- Cross-Connect
- High Density IP Access (HDIA)

The IP netmodel can be Full-IP or Policy-based. If Policy-based, it can run over a WAN or Ethernet link. The VWAN netmodel can be set in point-to-point or bridge (aggregate) mode over either a WAN or Ethernet link. Similarly, CopperVPN can be set in point-to-point or aggregate mode, but, unlike VWAN, it can specify the IP address of the far-end router or learn the router's IP address with inverse ARP.

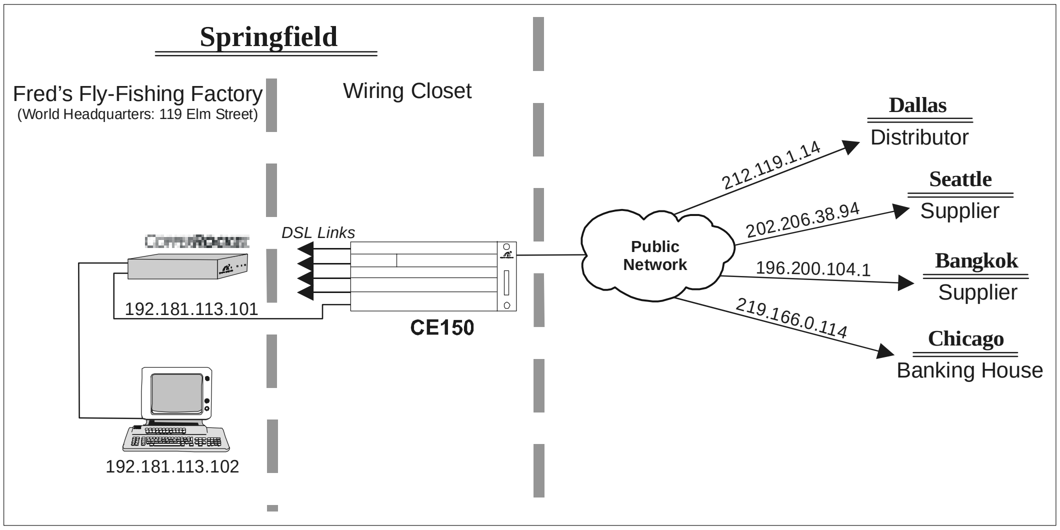
In contrast to VWAN, CopperVPN, and HDIA, the Cross-Connect netmodel only provides point-to-point connections, but it offers many encapsulation types and forwarding modes. Finally, the HDIA model can serve voice streams and data streams on the same port when they are separated into different VCs. For more details on the netmodels and their encapsulation types, see the cmIfaceTable in the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual.

N O T E

Configuring DSL interfaces for Voice-over-SDSL or for Multilink operation entails special requirements and precautions. Read and familiarize yourself with the detailed information provided in Chapter 7.

IP Netmodel

With this netmodel, the CE150 is able to route both inbound and outbound packets according to the IP address of their origination and destination devices. In the full IP netmodel (and also the IP policy variation), subscriber links appear as wide-area extensions of a remote LAN.



IP Network Model

Full IP Routing

The IP netmodel with full IP forwarding is one that most users know well. The NSP Central Office serves as a gateway to the net, and individual message packets are exchanged and routed over the public network based on their destination IP address. The diagram above describes the model.

1. Set the IP address, and other required parameters for each connected CPE. Use the PII to designate the interface to be configured.

```
CRAFT> set cmiface [1.4.1] netmodel=ip
ipaddr=192.181.113.101 netmask=255.255.255.0
encapsulationtype=rfc1483
Set Successful

CRAFT> get cmiface [1.4.1]
Group: cmIfaceTable
Instance: [1.2.1.0]
PII                  =  1.2.1.0
IfIndex              =  1.2.1.0
Name                 =  ""
GroupName            =  ""
AdditionalInfo       =  ""
NetModel             =  IP
IpAddr               =  192.181.113.101
NetMask              =  255.255.255.0
MacAddr              =  0.80.50.1.b3.ca
BurnedInMacAddr      =  0.80.50.1.b3.ca
FarEndAddr           =  0.0.0.0
DestPII              =  0.0.0.0
```

```
CMCPCompatible      =  No
EncapsulationType   =  rfc1483
FwdMode             =  Full-IP
Pix                 =  2
ServiceClass        =  None
```

2. The CE150 retrieves the MAC address from the CPE automatically each time a DSL link is established.
3. Remember also that *all* CE150 ports—Ethernet, WAN, or DSL—are referenced with a PII. For example PII 1.2.1 refers to the connector (Port 1) on the front of a DS3 WAN module.
4. Hosts connected to the CPE LAN must be configured with IP addresses on the subnet defined by the DSL port's IP address and netmask.
5. When you have completed the Initial Configuration of the CE150 and its interfaces, save the configuration data to a file:

   ```
   CRAFT> set cmsystem command=save
   ```

6. The file is saved on the appropriate volume (P:) as: \ce200\system\config.tgz (or config.txt). For more information, see *Saving Values* on page 108.
7. Use the ping command to verify connectivity.

   ```
   CRAFT> ping 192.181.113.102
   ```

8. Log out of the Command Line Interface by entering **`exit`**, **`quit`**, or **`logout`** at the prompt.
9. If you fail to log out, the system will perform an auto logout after approximately 15 minutes (same as Telnet).

Policy-based IP Routing

With this variation on the IP netmodel, the CE150 is able to route inbound packets according to the IP address of the origination device and outbound packets, regardless of IP address, to a designated interface on the CE150. Two types of interfaces exist:

- In IP Policy over a WAN interface, packets outbound from the CPE are routed to a destination WAN-port interface (usually an ATM VC or Frame Relay PVC) on the CE150.
- In IP Policy Over Ethernet, packets outbound from the CPE are routed to a specific LAN host device through the CE150 Ethernet port.

To configure interfaces for Policy-based IP routes and to establish WAN virtual circuits for IP-Policy, follow the procedures in the *IP Policy Over WAN* on page 119.

VWAN Netmodel

With this netmodel, subscriber links act as wide-area extensions of a remote LAN or other private network. Packets can be exchanged over an ATM or Frame Relay virtual circuit or, as MAC PDUs, over the CE150's Ethernet port. VWAN is a layer 2 LAN extension to an upstream router.

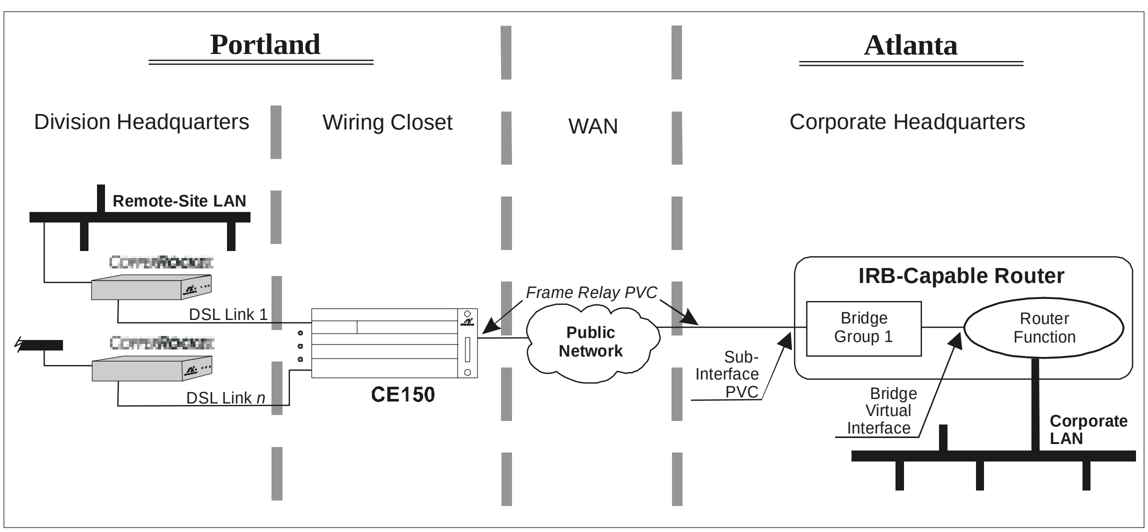
The VWAN netmodel requires an upstream IRB router, allowing the CE150 and the router together to function as a virtual bridge. Hosts on the LANs on the downstream side of the CPEs become remote or *wide-area* extensions of a private LAN on the upstream side of the router.

N O T E

Enabling VWAN in the CE150 *depends on a specific operating mode of certain routers called Integrated Router/Bridge or IRB. Routers without this capability may not be suitable for use in VWAN applications.*

The VWAN netmodel allows you to link a single DSL line to a single virtual bridge (WAN VC or Ethernet) in point-to-point mode, or aggregate many DSL links into a single virtual bridge, provided they all have the same DestPII; that is, provided they are served by a common circuit (WAN VC or Ethernet) on a common WAN (or Ethernet) interface linked to a common bridge group on a distant IRB router.

Each CE150 can support as many virtual bridge groups as it can virtual circuits, each connecting to a different far-end terminal point. Each bridge group, in such a case, must be associated with one DLCI or ATM Virtual Link number (or with the Ethernet port).



VWAN Network Model

When many DSL links comprise a virtual bridge group, the cmIface FwdMode object is VWAN BRIDGED. When only a single DSL link comprises a bridge group, the FwdMode object in the cmIface group is VWAN POINT-TO-POINT. When the Ethernet port is configured as the destination WAN interface, the FwdMode object is always VWAN BRIDGED, even for a single member group.

Configuration of VWAN links must be coordinated with the telecom/datacom management at the far end of the upstream link. An IRB-capable router with a compatible operating system (IOS) version must be installed at the far end.

The following procedure provides a model for end-to-end configuration of an SDSL subscriber for VWAN, using a Frame Relay WAN link. Configuration of the network port for VWAN over Ethernet appears in Chapter 5.

1. Configure the DSL interface for VWAN and specify the outbound facility to use.
2. Remember that, for a Frame Relay or ATM WAN interface, the DESTPII object includes the number (Frame Relay DLCI or ATM Virtual Link) that identifies the associated virtual circuit. For example:

```
CRAFT> set cmiface [1.4.1] netmodel=vwan
encapsulationtype=rfc1483 destpii=1.2.1.27
Set Successful

CRAFT> get cmiface [1.4.1]
Group: cmIfaceTable
Instance: [1.4.1.0]
PII                  = 1.4.1.0
IfIndex              = 1.4.1.0
Name                 = Zebra
GroupName            = X-ray
AdditionalInfo       = Comp account
NetModel             = VWAN
IpAddr               = 0.0.0.0
NetMask              = 0.0.0.0
MacAddr              = 0.60.58.1.0.36
BurnedInMacAddr      = 0.77.62.18.63.bc
FarEndAddr           = 0.0.0.0
DestPii              = 1.2.1.27
CMCPCompatible       = Yes
EncapsulationType    = rfc1483
FwdMode              = VWAN Point-to-Point
Pix                  = 72
Service Class        = D
```

3. The MAC address listed for the DSL port in VWAN mode is actually the address of the connected router at the far end of the WAN link. The CE150 still displays the Mac Address of the CPE, however, as the BurnedInMacAddr object.
4. Enable the WAN interface for Frame Relay.

```
CRAFT> set cmfrdlcmi [1.2.1] adminstate=enabled
Set Successful
```

5. The example here is a simplified one that does not specify LMI. To provide a link with LMI, use the State object in the standard frDlcmi group to specify the LMI formatting scheme. For more information on Frame Relay configuration, including LMI, see Chapter 6.

6. Activate the virtual circuit.

```
CRAFT> set frcircuit [1.2.1.27] state=active
Set Successful
```

7. Repeat step 1 for each port to be included in the Virtual Bridge group.

8. Optional: Configure the Virtual Bridge to block broadcast packets (such as IPX) to the wide area network.

```
CRAFT> set cmvbridge [1.2.1.27] option=ip-special
Set Successful
```

9. For more about cmVbridge, see the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual.

10. When you have completed the initial configuration of the CE150 and its interfaces, save the configuration data:

```
CRAFT> set cmsystem command = save
```

11. The file is saved on the appropriate volume (P:) as: \ce200\system\config.tgz (or config.txt). For more information, see *Saving Values* on page 108.

12. At the far end of the VWAN link (the main LAN site) configure the IRB-capable router to provide a bridge virtual interface (BVI).

13. For example, here is a typical config.txt file for a Cisco Systems IRB-capable router showing the relevant entries for VWAN (in this example, the DLCI number is 30):

```
!
interface Ethernet0
ip address 192.1.1.188 255.255.255.0
!
interface Serial0
 no ip address
 encapsulation frame-relay IETF
 no keepalive
 ignore-dcd
!
interface Serial0.1 point-to-point
 frame-relay interface-dlci 30
 bridge-group 1
!
interface Serial1
 no ip address
shutdown
!
interface BVI1
mac-address 0000.0c00.338c
ip address 206.41.72.1 255.255.255.0
no keepalive
!
bridge irb
 bridge 1 protocol ieee
 bridge 1 route ip
!
```

14. For additional information, please consult the documentation supplied by the manufacturer.

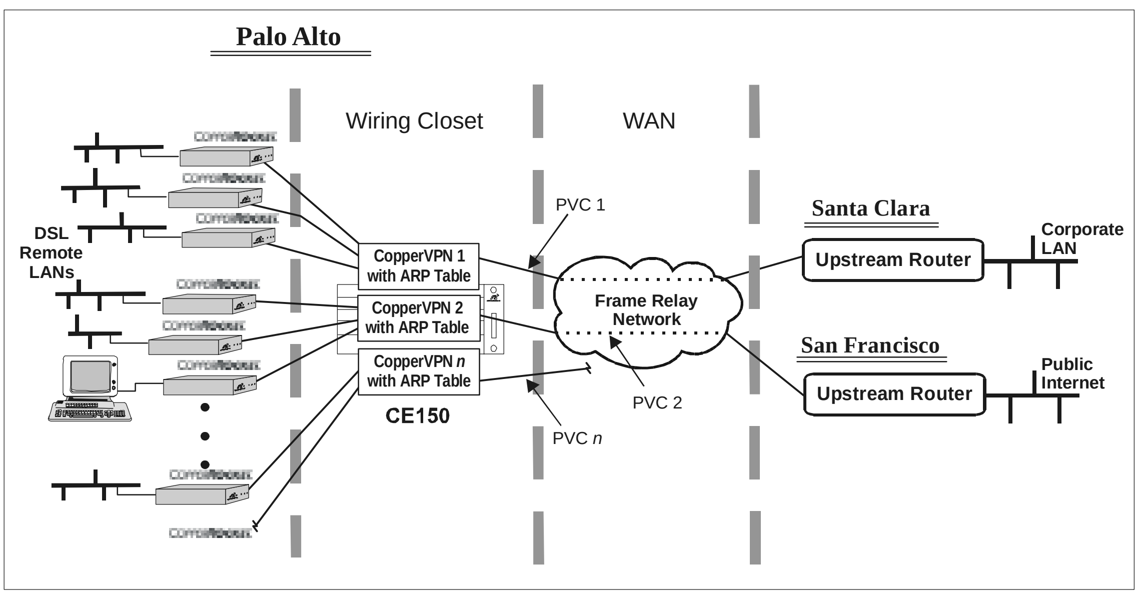
15. Log out of the Command Line Interface by entering **`exit`**, **`quit`**, or **`logout`** at the prompt.
16. If you fail to log out, the system will perform an auto logout after approximately 15 minutes (same as Telnet).

CopperVPN Netmodel

With this netmodel, either in static forwarding mode or auto learning mode, subscriber links act as wide-area extensions of a DSL remote LAN. Unlike the VWAN netmodel, CopperVPN does not rely on the upstream device for integration of routing and bridging functions. The CE150 learns the IP and MAC address and port number for every host supported by the CPEs.

CopperVPN is a layer 3 LAN extension to an upstream router. It simplifies the connection of DSL lines between end users and a remote LAN or other corporate network. With it, you can combine a number of DSL ports into a group within the CE150. You can send the data over a single virtual circuit across the network to a single destination, typically a router at an ISP or a Corporate site. Each CopperVPN PVC thus comprises a Virtual Private Network of DSL subscribers without the need for any special router capabilities such as IRB or compatible IOS versions.

For each CopperVPN group, the CE150 has a separate ARP table, which it uses to perform the IP-to-MAC address translation. Incoming IP PDUs from the upstream router are properly delivered to their intended DSLs and nowhere else. On the DSL LAN, the CE150 acts as a proxy for the upstream router, and replies to ARP requests from devices connected to the various DSL ports. Then, when the host on the DSL LAN responds with upstream transmission of MAC PDUs, the CE150 simply forwards them to the appropriate upstream router in IP PDU format. The CE150 obtains knowledge of the address of the upstream router in one of two ways: either from the static configuration (entry of the WAN PVC's cmIface FarEndAddr object), or from an inverse ARP request to the remote upstream device.



CopperVPN Network Model

Each CopperVPN group is separate and independent from the others, so groups can have overlapping IP addresses. This separation, together with the overall design of the CE150, also enhances security by preventing IP packets from crossing from one CopperVPN group to another. There is no restriction on the size of CopperVPN groups; a group may have one member or include every DSL port.

To establish a CopperVPN link, you must issue a series of set cmIface commands to the interfaces making up the CopperVPN group, to the DSL ports and the associated WAN VC.

1. For each DSL port to be aggregated, configure its NetModel, DestPII, and EncapsulationType objects as follows:

```
CRAFT> set cmiface [1.4.1] netmodel=coppervpn
destpii=1.4.1.23 encapsulationtype=rfc1483
Set Successful

CRAFT> get cmiface [1.4.1]
Group: cmIfaceTable
Instance: [1.4.1.0]
PII                  = 1.4.1.0
IfIndex              = 1.4.1.0
Name                 = Port Name
GroupName            = Group Name
AdditionalInfo       = Port Information
NetModel             = CopperVPN
IpAddr               = 0.0.0.0
NetMask              = 0.0.0.0
MacAddr              = ff.ff.ff.ff.ff.ff
BurnedInMacAddr      = 0.60.58.1.1.2c
FarEndAddr           = 0.0.0.0
DestPII              = 1.2.1.23
CMCPCompatible       = Yes
EncapsulationType    = rfc1483
FwdMode              = CopperVPN
Pix                  = 4
Service Class        = D
```

2. Identify the WAN VC on which to aggregate these DSL links by specifying its NetModel and EncapsulationType objects:

```
CRAFT> set cmiface [1.2.1.23] netmodel=coppervpn
encapsulationtype=rfc1490
Set Successful
```

If the WAN interface is ATM, use RFC1483 instead of RFC1490.

```
CRAFT> get cmiface [1.4.1.23]
Group: cmIfaceTable
Instance: [1.2.1.23]
PII                  = 1.2.1.23
IfIndex              = 1.2.1.23
Name                 = Port Name
GroupName            = Group Name
AdditionalInfo       = Port Info
NetModel             = CopperVPN
IpAddr               = 0.0.0.0
NetMask              = 0.0.0.0
MacAddr              = ff.ff.ff.ff.ff.ff
BurnedInMacAddr      = ff.ff.ff.ff.ff.ff
FarEndAddr           = 0.0.0.0
DestPII              = 0.0.0.0
CMCPCompatible       = No
EncapsulationType    = rfc1490
FwdMode              = CopperVPNauto
Pix                  = 4
Service Class        = D
```

3. If no IP address is specified for the VC's FarEndAddr (it is 0.0.0.0), the system will use inverse ARP to get the upstream IP address and FwdMode will display `CopperVPNauto`. If you set FarEndAddr, the static setting overrides the inverse ARP function. The FwdMode object displays `CopperVPN`, and outbound packets go to the location specified. Since CopperVPN DSL interfaces specify a static endpoint (the DestPII), their FwdMode is always CopperVPN.
4. At the far end of the CopperVPN link (the upstream router), configure the router to specify the VC.
5. In the sample config.txt file for a Cisco Systems router, the settings are the same as for the VWAN netmodel, but without IRB functionality; IP addresses are assigned on the subinterface in its place:

```
version 11.2
no service udp-small-servers
no service tcp-small-servers
!
hostname MegaFonsInc
!
enable password admin
!
frame-relay switching
!
interface Ethernet0
 ip address 192.168.1.1 255.255.255.0
!
interface Serial0
 no ip address
 encapsulation frame-relay IETF
 no keepalive
 ignore-dcd
 frame-relay intf-type dce
!
interface Serial0.1 point-to-point
 frame-relay interface-dlci 22
 ip address 192.168.2.1 255.255.255.0
!
interface Serial0.2 point-to-point
 frame-relay interface-dlci 23
 ip address 192.168.3.1 255.255.255.0
!
interface Serial0.3 point-to-point
 frame-relay interface-dlci 24
 ip address 192.168.4.1 255.255.255.0
!
interface Serial1
 no ip address
 encapsulation frame-relay IETF
 no keepalive
 ignore-dcd
 shutdown
!
no ip classless
ip route 0.0.0.0 0.0.0.0 192.168.1.254
!
snmp-server community public RO
!
line con 0
line aux 0
line vty 0
line vty 1 4
```

Cross-Connect Netmodel

The Cross-Connect netmodel allows conversion of disparate frame formats between DSL and WAN interfaces. Under this netmodel, different forwarding modes support multiple encapsulation types as well as translation between different encapsulation types. For example, under the VC-VC Payload forwarding mode, a frame encapsulation on the DSL interface can be converted into an ATM encapsulation on the WAN interface.

There are two types of Cross-Connect forwarding modes, determined by the DSL port encapsulation: *per-VC* forwarding modes and *per-port* forwarding modes. P*er-VC* forwarding modes result from a DSL encapsulation that allows multiple VCs per DSL port. P*er-port* forwarding modes result from a DSL encapsulation that does not support VCs.

Per-VC Forwarding Modes

All Cross-Connect per-VC encapsulations support up to 64 PVCs on the DSL port multiplexed to a single WAN VC. Each DSL VC inherits its encapsulation from the DSL port. DLCI numbers identifying DSL VCs obey the same rules as they do on WAN VCs: numbers must be between 17 and 991, and they do not have to be sequential. However, the numbers 16, 528, and 529 cannot be used.

In general, the DSL VC's forwarding mode is determined by the WAN VC. If the WAN encapsulation is not specified (None), packet forwarding is performed in VC-VC Payload forwarding mode. If the DSL and WAN encapsulations are the same, packet forwarding is performed in a *transparent* mode. If the DSL and WAN encapsulations differ, packet forwarding is performed in a *translation* mode.

Since the cmSubIface table has no FwdMode object, the Cross-Connect forwarding mode is found in the cmIfaceTable entry for the WAN VC. The forwarding mode for the DSL port is per-VC, since each DSL VC has its own FwdMode.

LMI is available on DSL links with per-VC forwarding modes. When LMI is in force, the CE150 acts as a DCE, and the CPE acts as a DTE. For more information, see the discussion on the cmIface and cmSubIface object groups in the CopperEdge 150 CopperCraft Reference and MIB Definitions manual. Also see the discussion of encapsulation in Appendix A.

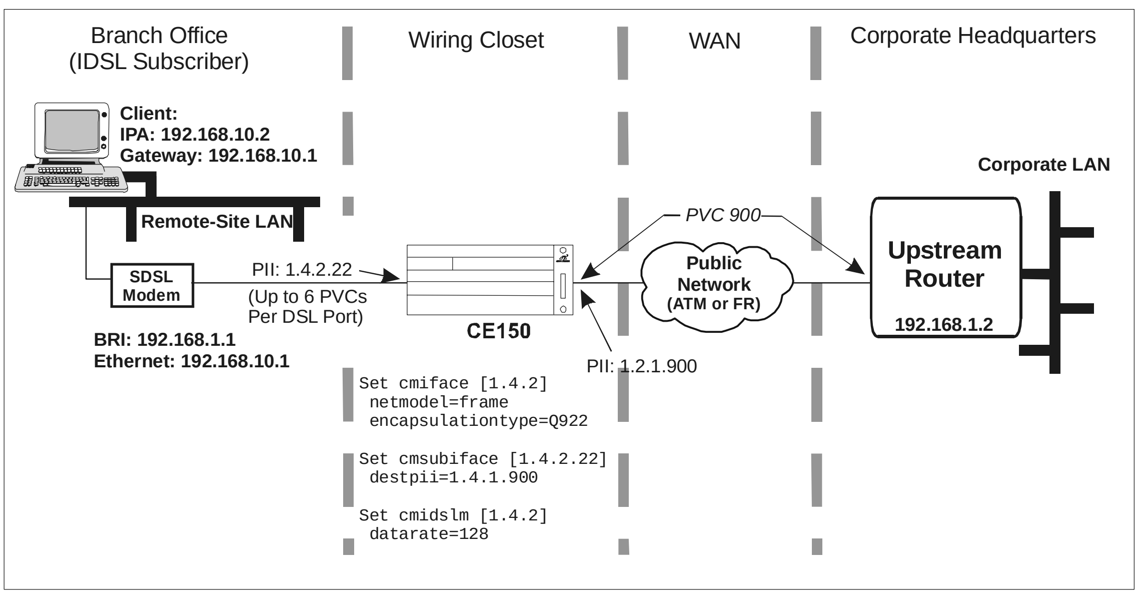
Encapsulations on DSL and WAN VCs for Per-VC Forwarding

DSL Encapsulation	Line Card Type	WAN Encapsulation	Forwarding Mode
Q922-1490, Q922	Frame	None	VC-VC-payload (transparent)
Q922-1490, Q922	Frame	FRF5	FRF5 (translation)
Q922-1490	Frame	RFC-1483	FRF8-1490-1483 (translation)
RFC-1973	Frame	RFC-1973	PPP-Transparent
RFC-1973	Frame	RFC-2364 null	PPP-Translation
RFC-1973	Frame	RFC-2364 LLC	PPP-Translation
RFC-1973	Frame	None	VC-VC-Payload

VC-VC Payload Forwarding Mode

The VC-VC payload forwarding mode allows third-party CPEs to use encapsulation formats other than RFC-1483 so they can function as terminals on a remote LAN. In such a configuration, as the following graphic shows, the WAN interface and backbone can be either Frame Relay or ATM. DSL VCs configured for VC-VC payload using the Cross-Connect netmodel generally use the Q922-1490 encapsulation type. This setting, used *only* on DSL VCs, specifies:

- The DSL physical port encapsulation is Q.922, thus allowing Frame-Relay or FUNI VCs on it.
- The encapsulation on every VC using that port is RFC-1490.



VC-VC Payload Fowarding Model

FRF.5 Forwarding Mode

This translation mode allows a DSL port on the CE150 to receive Q.922 HDLC frames, transform them into ATM AAL5 PDUs, and send them to an ATM WAN interface for transmission across an ATM network. At the other end of the network, a reverse translation allows the Q.922 HDLC frames to be recovered.

To configure this mode, set the netmodel and encapsulation type on the DSL and WAN interfaces:

```
CRAFT> set cmiface [1.2.1.100] netmodel=cross-connect
encapsulationtype=frf5
```

```
CRAFT> set cmiface [1.5.1] netmodel=cross-connect
encapsulationtype=q922
```

```
CRAFT> set cmsubiface [1.5.1.15] destpii=1.2.1.100
```

FRF.8 Translation and Transparent Forwarding Modes

This translation/forwarding mode is for Q.922 frames being sent to ATM VCs on an ATM network where they terminate. The CE150 decides to use translation mode or transparent mode based on the encapsulations set on the DSL VC and WAN VC. (In the Cross-Connect netmodel, DSL VC encapsulation is implied by the DSL port encapsulation.)

For instance, if the two encapsulations are the same (or if the WAN VC encapsulation is None), Transparent mode is used. If they differ, the Translation mode is used. Since DSL encapsulation applies to an entire DSL port, all DSL VCs that require translation must use the same encapsulation. DSL encapsulation is not relevant or used on DSL VCs that are not translated (i.e., whose cross-connected WAN VC does not specify a translation).

To configure two data VCs on one DSL port, and map them to two WAN VCs where one has an encapsulation type of RFC-1483 and the other has an encapsulation type of None, enter the following commands:

```
CRAFT> set cmiface [1.2.1.1] netmodel=cross-connect
encapsulationtype=rfc1483 destpii=1.4.1.19
```

```
CRAFT> set cmiface [1.2.1.2] netmodel=cross-connect
encapsulationtype=none destpii=1.4.1.20
```

```
CRAFT> set cmiface [1.4.1] netmodel=cross-connect
encapsulationtype=q.922
```

```
CRAFT> set cmsubiface [1.4.1.19] destpii=1.2.1.1
```

```
CRAFT> set cmsubiface [1.4.1.20] destpii=1.2.1.2
```

After configuration, WAN VC 1.3.1.1 has a forwarding mode of FRF8-1490-1483, while WAN VC 1.3.1.2 has a forwarding mode of VC-VC-payload.

Not all translations described by the FRF.8.1 specification are implemented. Translations are implemented for Frame Relay PDUs with:

- NLPID=0x80, which include but are not limited to Bridged PDUs (802.3 protocol) and Other routed PDUs (AppleTalk data, AppleTalk ARP, IPX, Inverse ARP but not ARP)
- NLPID=0xCC, which are Routed IP Version 4 PDUs
- NLPID=0x08, Connection oriented protocols (Q.933/Q.2931 NLPID) PDUs that include but are not limited to SNA

Packets that do not fall into these categories will be discarded.

PPP Translation and Transparent Forwarding Modes

The PPP-translation forwarding mode supports RFC-1973 to RFC-2364 interworking. PPP-transparent forwarding mode supports RFC-1973 to RFC-1973 interworking. At present, RFC-2364 to RFC-1973 and RFC-2364 to RFC-2364 interworking are not supported.

RFC-2364 actually specifies two encapsulations, the Null (VC Multiplexing) and the Logical Link Control (LLC) encapsulations. ATM based line cards and WAN interfaces in the CE150 support both RFC-2364 encapsulations.

To convert an RFC-1973 frame to RFC-2364, the flag and Q.922 header are stripped away. For the RFC-2364 Null encapsulation, Control and NLPID are removed as well. For the RFC-2364 LLC encapsulation, an LLC header is prepended to the frame and the NLPID is retained. With both RFC-2364 encapsulations, an optional pad and a CPCS-PDU trailer are appended to the frame at the ATM driver level. The process is reversed to convert an RFC-2364 frame to RFC-1973.

The following frame-based line cards and WAN interfaces in the CE150 support RFC-1973 encapsulation:

- SDSL
- T1
- DS3 Frame
- Quad T1

The following ATM-based line cards and WAN interfaces in the CE150 support RFC-2364 encapsulation:

- G.lite
- G.dmt
- DS3 ATM

RFC-1973 is configured on the port level for DSL links. Then DSL VCs are configured with cmSubIface entries. The DSL VCs assume the DSL port encapsulation.

To configure DSL RFC-1973 to WAN RFC-2364-null:

1. Set the DSL port netmodel and encapsulation:

```
CRAFT> set cmiface [1.5.1] netmod=cross-connect
encapsulationtype=rfc1973
```

2. Create the cmSubIface row and set the destination PII:

```
CRAFT> set cmsubiface [1.5.1.20] destpii=1.2.1.100
```

3. Set WAN VC encapsulation:

```
CRAFT> set cmiface [1.2.1.100] encapsulationtype=
rfc2364-null
```

4. Create a cmAtmVcl row for the WAN VC:

```
CRAFT> set cmatmvcl [1.2.1.100] vpi=1 vci=121
adminstatus=up
```

After configuration, WAN VC 1.3.1.100 has a forwarding mode of PPP-translation. As with other per-VC encapsulations, the DSL VC's forwarding mode is the same as that of the cross-connected WAN VC.

To configure DSL RFC-1973 to WAN RFC-1973:

1. Set the netmodel and encapsulation on the DSL port:

```
CRAFT> set cmiface [1.5.1] netmodel=cross-connect
encapsulationtype=rfc1973
```

2. Create the cmSubIface row and set the WAN destination PII:

```
CRAFT> set cmsubiface [1.5.1.20] destpii=1.2.1.100
```

3. Set the WAN VC encapsulation:

```
CRAFT> set cmiface [1.2.1.100] encapsulationtype=
rfc1973
```

After configuration, both WAN VC 1.3.1.100 and DSL VC 1.6.1.20 have a forwarding mode of PPP-transparent.

Per-Port Forwarding Modes

The Cross-Connect netmodel per-port forwarding modes result from a DSL encapsulation that defines the DSL port as a single logical interface that does not support VCs.

Encapsulations on DSL and WAN VCs for Per-Port Forwarding

DSL Encapsulation	Line Card Type	WAN Encapsulation	Forwarding Mode
PPP-HDLC	Frame	RFC-1973	PPP-HDLC-1973
PPP-HDLC	Frame	None	HDLC-VC-payload

PPP-HDLC Forwarding Modes

The Cross-Connect netmodel supports PPP-HDLC encapsulation, which is valid only for DSL ports, not for WAN VCs. The forwarding mode in the previous table applies to both the DSL port and the WAN VC.

PPP-HDLC (RFC-1662) is a point-to-point protocol that allows a DSL port on the CE150 to receive PPP-HDLC frames. The CE150 can transform them into RFC-1973 frames, and then send them to a WAN VC. Different ports carrying PPP-HDLC are mapped to VCs on the same WAN interface. The operation is called *Point-to-Point Protocol Multiplexing*.

To configure the PPP-HDLC-1973 forwarding mode, set the netmodel and encapsulation on the DSL and WAN interfaces:

1. Set the DSL port's netmodel and encapsulation, and specify the WAN VC as the destination PII:

   ```
   CRAFT> set cmiface [1.4.1] netmodel=cross-connect
   encapsulationtype=ppp-hdlc destpii=1.2.1.100
   ```

2. Set the netmodel and encapsulation for the WAN VC:

   ```
   CRAFT> set cmiface [1.2.1.100) netmodel=cross-
   connect encapsulationtype=rfc1973
   ```

After configuration, both WAN VC 1.3.1.100 and DSP port 1.6.1 have a forwarding mode of PPP-HDLC-1973.

To configure the HDLC-VC-payload forwarding mode, set the netmodel and encapsulation on the DSL and WAN interfaces:

1. Set the DSL port's netmodel and encapsulation, and specify the WAN VC as the destination PII:

   ```
   CRAFT> set cmiface [1.4.1] netmodel=cross-connect
   encapsulationtype=ppp-hdlc destpii=1.2.1.100
   ```

2. Set the netmodel and no encapsulation for the WAN VC:

   ```
   CRAFT> set cmiface [1.2.1.100) netmodel=cross-
   connect encapsulationtype=none
   ```

After configuration, both WAN VC 1.3.1.100 and DSP port 1.6.1 have a forwarding mode of HDLC-VC-payload.

For more information about the Cross-Connect netmodel, refer to Appendix A, *Data Encapsulation*.

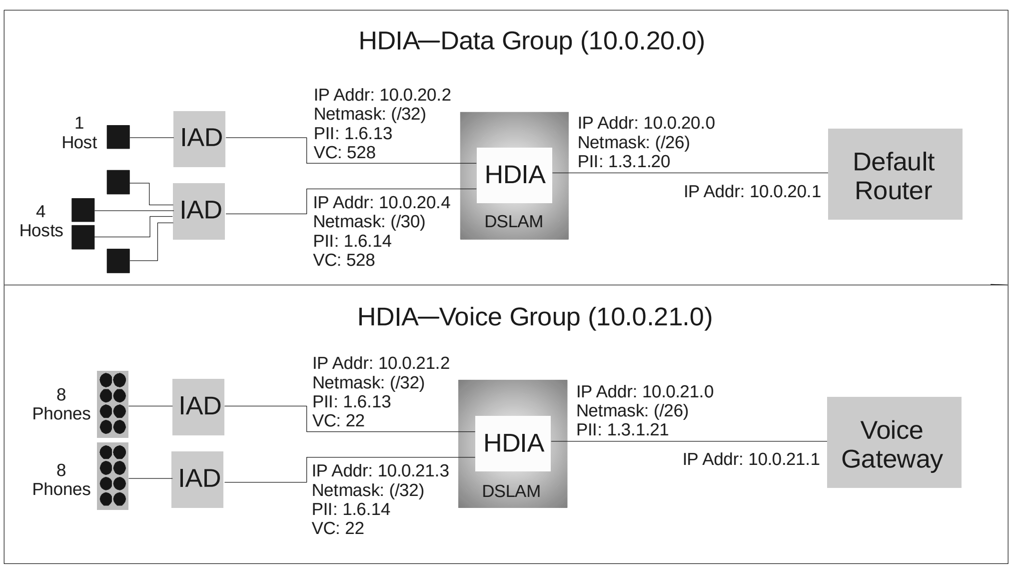
High Density Internet Access (HDIA) Netmodel

With this netmodel, the CE150 is able to provide economical IP addressing for hosts. It allows the mapping of all data VCs on different DSL ports to a single data VC on a WAN port and mapping of all high priority VCs (always VC 22) on different DSL ports to a single VC on a WAN port. High priority VCs are for delay-sensitive traffic, like voice.

HDIA, unlike VWAN or CopperVPN in their aggregate mode, permits peer-to-peer communications for users on voice VCs.

With HDIA, you must set base IP addresses and address ranges (or net masks) for voice and data subnets when configuring the circuits on WAN ports. Also, you must specify base IP addresses and address ranges (or net masks) for IADs and hosts when configuring circuits on DSL ports. For more detailed instructions, see the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual. Here are examples of configurations for the data and voice VCs on both the DSL and WAN interfaces.

```
CRAFT> set cmiface [1.2.1.20] netmodel=hdia ipaddr=10.0.20.0
netmask=255.255.255.192 encapsulationtype=rfc1483
farendaddr=10.0.20.1

CRAFT> set cmiface [1.2.1.21] netmodel=hdia ipaddr=10.0.21.0
netmask=255.255.255.192 encapsulationtype=rfc1483
farendaddr=10.0.21.1

CRAFT> set cmiface [1.5.13] netmodel=hdia ipaddr=10.0.20.2
netmask=255.255.255.255 destpii=1.2.1.20
encapsulationtype=rfc1483

CRAFT> set cmiface [1.5.13.22] netmodel=hdia
ipaddr=10.0.21.2 netmask=255.255.255.255 destpii=1.2.1.21
encapsulationtype=ip-1490
```



Configuring Data and Voice VCs

Configuring the CE150 for DHCP Functionality

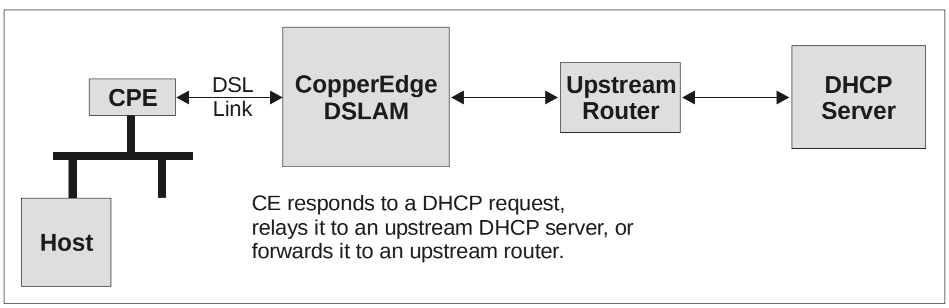
For specific ports or VCs, you can configure the CE150 to process DHCP requests in one of the following modes:

- *DHCP Server*—The CE150 responds to a DHCP request from a CMCP (Copper Mountain Compatible Protocol) CPE or a non-CMCP CPE, sending the IP address entered in the cmDHCPTable for that DSL PII.
- *DHCP Relay Agent*—The CE150 sends the DHCP request from any CPE or premise LAN host to an upstream DHCP server. The netmodel must be IP.
- *DHCP Forwarding Agent*—The CE150 sends the DHCP request from any CPE or premise LAN host to an upstream DHCP server. The netmodel must be Cross-Connect, VWAN, HDIA, or CopperVPN.

NOTE

Configured as a DHCP server, the CE150 cannot distinguish between non-CMCP CPEs and their hosts on the premise LAN. It can assign only one IP address from the cmDHCPTable, sending it to both non-CMCP CPEs and their hosts. Therefore, configure the CE150 to relay or forward the requests to an upstream DHCP server, which can respond with different IP addresses for both the CPEs and their hosts.

The CE150's DHCP functionality, based on RFC-1542, gives service providers the option of having their own DHCP servers provide IP configuration parameters for CPEs and hosts on the premise LAN.



DHCP Functionality in the CE150

The CE150 must use some form of client/circuit identifier when sending DHCP requests to an upstream DHCP server if that server is to be able to uniquely identify the clients (CPEs and their hosts) sending the requests. The CE150 assigns a client/circuit identifier for CPEs as follows:

- A unique circuit identifier is automatically entered in the CircuitID object in the cmDHCPTable. The default is the CE150's system name plus the DSL PII. You can change it to a different alphanumeric identifier, but it must start with an alpha character.
- The CircuitID is inserted in the Relay Agent Information (Option 82) and Client Identifier (Option 61) fields in the DHCP request message.

NOTE

An upstream DHCP server uses Option 82, Relay Agent Information, to identify which DSL port the DHCP request originated on. It uses Option 61, Client Identifier, to distinguish between requests originating from a CPE and requests originating from a host. If Option 61 is present, the request originated from a CPE; if it is not present, the request originated from a host.

The CE150's DHCP functionality depends on which netmodel is being used and whether the CPE is a CMCP CPE.

Summary of the CE150's DHCP Functionality

DHCP Function	Network Model	CMCP CPE	Host with CMCP CPE	Non-CMCP CPE or Host
CE150 as a Server	IP	Response	Relay	Response[1]
	CopperVPN	Response	Forward	Response[1]
	VWAN	Response	Forward	Response[1]
	HDIA	Response	Forward	Response[1]
	Cross-Connect	Response	Forward	Response[1]
CE150 as a Relay Agent	IP	Relay[2]	Relay[3]	Relay[3]
CE150 as a Forwarding Agent	CopperVPN	Forward[2]	Forward[3]	Forward[3]
	VWAN	Forward[2]	Forward[3]	Forward[3]
	HDIA	Forward[2]	Forward[3]	Forward[3]
	Cross-Connect	Forward[2]	Forward[3]	Forward[3]

1. The CE150 cannot distinguish between non-CMCP CPEs and their hosts on the premise LAN. It can only assign one IP address, which comes from the cmDHCPTable. The CE150 sends that IP address to both the non-CMCP CPEs and their hosts. Therefore, configure the CE150 to either Relay or Forward these requests to an upstream DHCP server, which can assign different IP addresses for both.
2. Both the Relay Agent Information and Client Identifier are inserted in the request.
3. Only the Relay Agent Information is inserted in the request.

The following sections describe the CE150's DHCP functionality, both upstream and downstream.

Upstream DHCP Processing

In the upstream direction, the CE150 can function as a DHCP Server, Relay Agent, or Forwarding Agent for a DSL PII, depending on the netmodel.

You use the cmDHCPTable to configure DHCP functionality for the DSL PII. If there is no entry in the cmDHCPTable for the DSL PII, the CE150's default mode is to pass the DHCP request to the upstream router or DHCP server without inserting the Relay Agent Information or Client Identifier in the message. See Chapter 3 of the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual for information about the cmDHCPTable.

IP Netmodel

When the netmodel is IP, you can configure the CE150 to act as a DHCP Server or as a DHCP Relay Agent for a DSL port or VC.

The CE150 as a DHCP Server

For CMCP CPEs:

- The CE150 responds to DHCP requests from CMCP CPEs for which there is an entry in the cmDHCPTable.
- The CE150 generates a response based on the configured objects in the cmDHCPTable and sends the response back to the DSL PII that it arrived on.

For hosts on the premise LAN:

- The CE150 relays the DHCP requests to an upstream DHCP server. .
- The CE150 must know the IP address of the upstream DHCP server; you enter this address in the cmDHCPTable.
- The CE150 must also have an IP address assigned to itself, which the upstream DHCP server will use to route the DHCP responses back to the CE150.

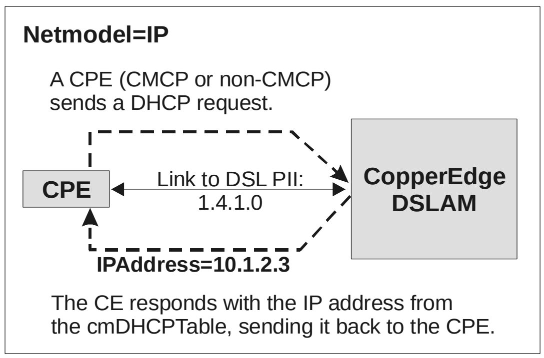
For non-CMCP CPEs and hosts on the premise LAN:

- The CE150 responds to DHCP requests from non-CMCP CPEs for which there is an entry in the cmDHCPTable.
- However, the CE150 cannot differentiate between DHCP requests coming from non-CMCP CPEs or hosts on the premise LAN. It sends the same response for both.
- Therefore, for non-CMCP CPEs, configure the CE150 as a Relay Agent instead of as a DHCP Server.

To configure the CE150 to act as a DHCP Server for a port or VC:

In cmDHCPTable, enter the upstream DHCP server's IP address in the ServerIPAddr object and set the Function object to DHCPRespond. You can leave the CircuitID object at the default (the system name and the PII of the port or VC).

```
CRAFT> set cmdhcp [1.4.1.0] function=dhcprespond
serveripaddr=204.20.20.30
Set Successful

CRAFT> get cmdhcp [1.4.1.0]
Group: cmDHCPTable
Instance: [1.4.1.0]
PII                        = 1.4.1.0
RowStatus                  = Active
IpAddress                  = 10.1.2.3
NetMask                    = 255.255.255.255
DefaultRoute               = 204.20.20.10
DNSServer                  = 204.20.20.20
Function                   = DHCPRespond
ServerIPAddr               = 204.20.20.30
CircuitID                  = CMTN-1.4.1.0
```



The CE150 as a DHCP Server

The CE150 as a DHCP Relay Agent

The CE150 relays DHCP requests from CPEs (both CMCP and non-CMCP) and hosts on the premise LAN to an upstream DHCP server.

To configure the CE150 to act as a DHCP Relay Agent for a port or VC:

In cmDHCPTable, enter the upstream DHCP server's IP address and set the Function object to DHCPRelay. You can leave the CircuitID object at the default (the system name and the PII of the port or VC).

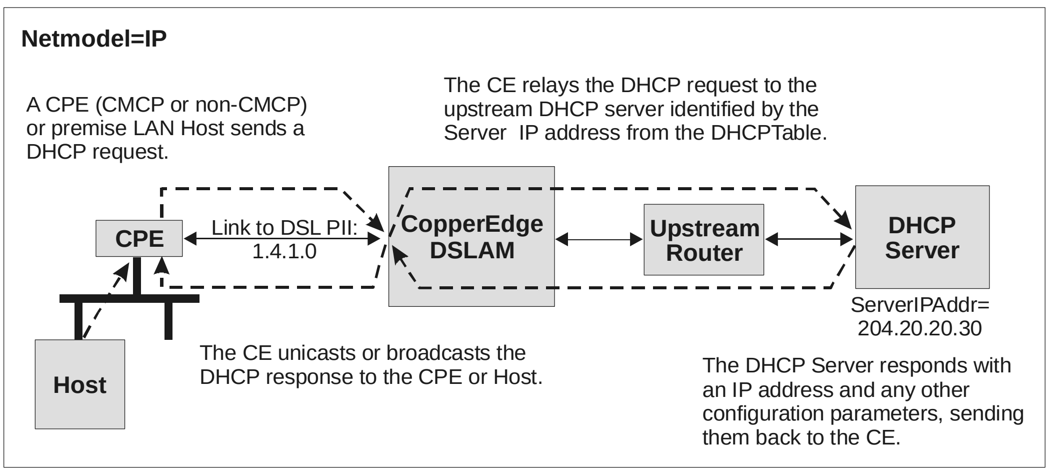
```
CRAFT> set cmdhcp [1.4.1.0] function=dhcprelay
serveripaddr=204.20.20.30
Set Successful

CRAFT> get cmdhcp [1.4.1.0]
Group: cmDHCPTable
Instance: [1.4.1.0]
PII                        = 1.4.1.0
```

```
RowStatus          = Active
IpAddress          = 10.1.2.3
NetMask            = 255.255.255.255
DefaultRoute       = 204.20.20.10
DNSServer          = 204.20.20.20
Function           = DHCPRelay
ServerIPAddr       = 204.20.20.30
CircuitID          = CMTN-1.4.1.0
```

The CE150 performs the following functions when it receives a DHCP request:

1. The CE150 inserts the IP address of the DSL PII, over which the request was received, in the DHCP request. This IP address will be used by the upstream DHCP server to send the DHCP response back to the CE150.
2. For a CMCP CPE, the CE150 modifies the DHCP request by inserting the CircuitID from the cmDHCPTable into the Relay Agent Information and Client Identifier fields in the DHCP request message.

   For a non-CMCP CPE or a host on the premise LAN, the CE150 modifies the DHCP request by inserting the CircuitID only into the Relay Agent Information field.
3. The CE150 sends the request upstream as a unicast IP packet destined to the DHCP Server IP address configured in the cmDHCPTable.



The CE150 as a DHCP Relay Agent

Pass Through Mode

If the CE150 receives a DHCP request and there is no entry in the cmDHCPTable for that DSL PII, the CE150 passes the request to the upstream router or DHCP server. The DHCP request is not modified (the Relay Agent Information and Client Identifier fields are not inserted in the message).

When the netmodel is IP, the CPE or premise LAN host *cannot* obtain an IP address from the upstream DHCP server. The CE150 simply drops the response it receives from the upstream DHCP server.

Cross-Connect, HDIA, VWAN, and CopperVPN Netmodels

When the netmodel is Cross-Connect, HDIA, VWAN, or Copper-VPN, you can configure the CE150 to act as a DHCP Server or as a DHCP Forwarding Agent for a port or VC.

The CE150 as a DHCP Server

For CMCP CPEs:

- The CE150 responds to DHCP requests from CMCP CPEs for which there is an entry in the cmDHCPTable.
- The CE150 generates a response based on the configured objects in the cmDHCPTable and sends the response back to the DSL port that it arrived on.

For hosts on the premise LAN:

- The CE150 forwards the DHCP requests to an upstream DHCP router, which is responsible for sending the request to the appropriate DHCP server.
- The Forwarding function is described on page 85.

For non-CMCP CPEs and hosts on the premise LAN:

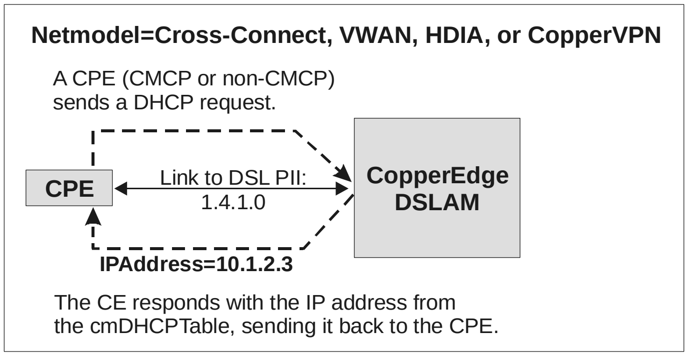
- The CE150 responds to DHCP requests from non-CMCP CPEs for which there is an entry in the cmDHCPTable.
- However, the CE150 cannot differentiate between DHCP requests coming from non-CMCP CPEs or hosts on the premise LAN. It sends the same response for both.
- Therefore, for non-CMCP CPEs, configure the CE150 as a Forwarding Agent instead of as a DHCP Server. See page 85.

To configure the CE150 to act as a DHCP Server for a port or VC:

In cmDHCPTable, set the Function object to DHCPRespond. You can leave the CircuitID object at the default (the system name and the PII of the port or VC).

```
CRAFT> set cmdhcp [1.4.1.0] function=dhcprespond
Set Successful

CRAFT> get cmdhcp [1.4.1.0]
Group: cmDHCPTable
Instance: [1.4.1.0]
PII                        = 1.4.1.0
RowStatus                  = Active
IpAddress                  = 10.1.2.3
NetMask                    = 255.255.255.255
DefaultRoute               = 204.20.20.10
```

```
DNSServer                    = 204.20.20.20
Function                     = DHCPRespond
ServerIPAddr                 = 0.0.0.0
CircuitID                    = CMTN-1.4.1.0
```



The CE150 as a DHCP Server

The CE150 as a DHCP Forwarding Agent

The CE150 forwards DHCP requests from CPEs (both CMCP and non-CMCP) and hosts on the premise LAN to an upstream WAN VC or router.

To configure the CE150 to act as a DHCP Forwarding Agent for a port or VC:

In cmDHCPTable, set the Function to DHCPForward. You can leave the CircuitID object at the default (the system name and the PII of the port or VC).

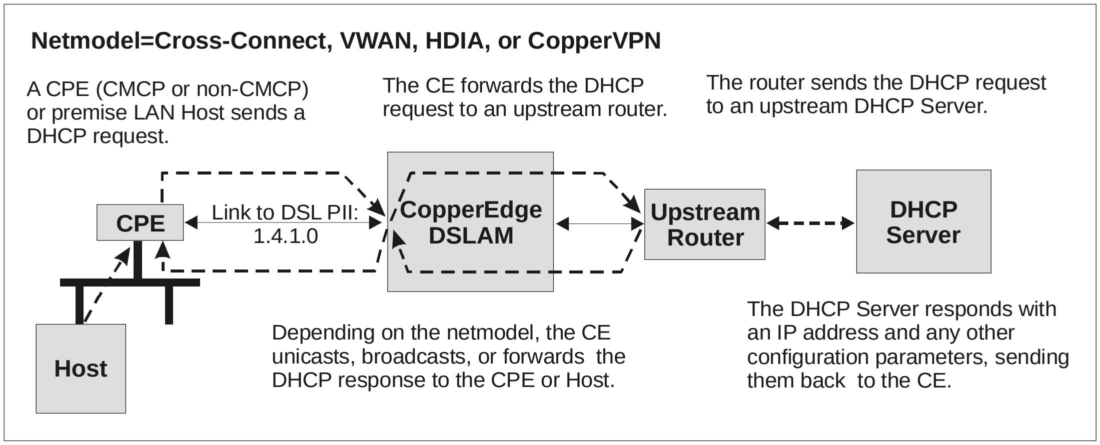
```
CRAFT> set cmdhcp [1.4.1.0] function=dhcpforward
serveripaddr=204.20.20.30
Set Successful

CRAFT> get cmdhcp [1.4.1.0]
Group: cmDHCPTable
Instance: [1.4.1.0]
PII                          = 1.4.1.0
RowStatus                    = Active
IpAddress                    = 10.1.2.3
NetMask                      = 255.255.255.255
DefaultRoute                 = 204.20.20.10
DNSServer                    = 204.20.20.30
Function                     = DHCPForward
ServerIPAddr                 = 0.0.0.0
CircuitID                    = CMTN-1.4.1.0
```

The CE150 performs the following functions when it receives a DHCP request:

1. For a CMCP CPE, the CE150 modifies the DHCP request by inserting the CircuitID from the cmDHCPTable into the Relay Agent Information and Client Identifier fields in the DHCP request message.

   For a non-CMCP CPE or a host on the premise LAN, the CE150 modifies the DHCP request by inserting the CircuitID only into the Relay Agent Information field.

2. The CE150 forwards the request to the upstream WAN VC or router, which is responsible for responding to or relaying the request to the upstream DHCP server.



The CE150 as a DHCP Forwarding Agent

Pass Through Mode

If the CE150 receives a DHCP request and there is no entry in the cmDHCPTable for that DSL PII, the CE150 passes the request to the upstream router or DHCP server. The DHCP request is not modified (the Relay Agent Information and Client Identifier fields are not inserted in the message).

When the netmodel is Cross-Connect, VWAN, HDIA, or CopperVPN, the CPE or premise LAN host can obtain an IP address from the upstream DHCP server. There is enough information in the message for the CE150 to forward the response to the correct DSL PII and ultimately to the correct CPE or premise LAN host.

Downstream DHCP Processing

In the downstream direction, the following DHCP processing takes place for DHCP Response messages for the various netmodels. The CE150 will drop any DHCP requests coming downstream from a WAN VC or an Ethernet port.

IP Netmodel

The DHCP response is routed to the DSL PII based on the IP address of that DSL PII and the assigned Client IP address of the received DHCP response. The DHCP response is either broadcast downstream on the DSL PII or sent as a unicast packet based on the DHCP broadcast flag.

Cross-Connect and VWAN Netmodels

The downstream DHCP responses are simply forwarded to the appropriate DSL PIIs based on the provisioning configured in the CE150. No additional processing is required.

HDIA Netmodel

If the received DHCP response is a broadcast IP packet, this packet is sent to the DSL PII within whose range the assigned client IP address in the DHCP response message falls. The response is sent as a broadcast IP message, and the DHCP client identifies its response based on the MAC Address in the DHCP response (which was previously sent in the DHCP request).

If the received DHCP response is an IP unicast packet, the CE150 sends this packet to the DSL PII within whose range the assigned client IP address in the DHCP response falls. The response is sent as a unicast packet and uses the MAC Address in the DHCP response as the unicast MAC address. Therefore, only the correct DHCP client will be able to receive the response.

CopperVPN Netmodel

If the received DHCP response is a broadcast IP packet, it is flooded to all members of the CopperVPN group as a broadcast, and the DHCP client identifies its response based on the MAC Address in the DHCP response.

If the received DHCP response is an IP unicast packet, the CE150 replicates this packet to all DSL PIIs in the CopperVPN group as a unicast packet, and uses the MAC Address in the DHCP response as the unicast MAC address. Therefore, only the correct DHCP client will be able to receive the response.

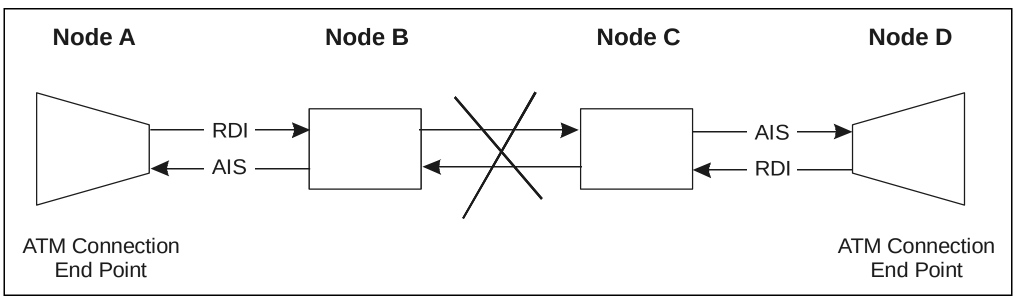
OAM Fault Management for ATM WAN Links

The OAM fault management function on the CE150 allows system administrators to monitor basic conditions for transmitting and receiving on end-to-end ATM WAN links, made up of many segments. It also allows them to locate a break on one of the segments.

Transmitting/Receiving Fault Messages

When a fault occurs on upstream or downstream segments, OAM fault management software installed on switching or routing devices on the segments sends out Alarm Indication Signal (AIS) messages. The messages travel to devices (such as DSLAMs and routers) at opposite ends of the link. In response, the DSLAM and routers send out Remote Defect Indication (RDI) messages.

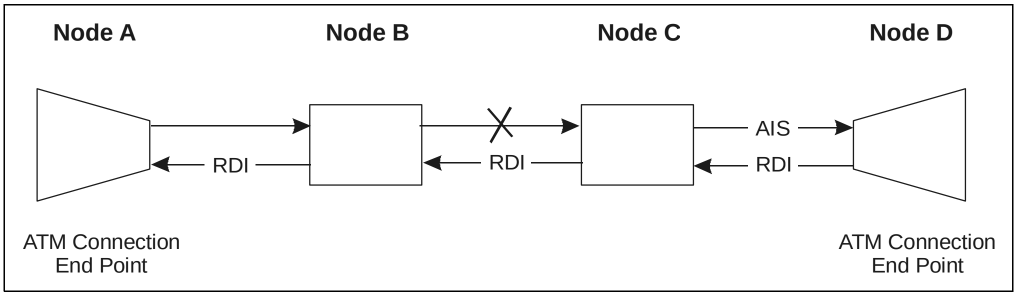
The diagrams below show two possible scenarios: first, a fault on both the upstream and downstream parts of a connection; second, a fault on the upstream part of a connection.



A Break on the Upstream and Downstream Connections

In the diagram above, the devices at Node B and Node C sense the fault between them. The device at Node B sends an AIS message to the device at the endpoint on Node A. Similarly, the device at node C sends an AIS message to the device on the endpoint at Node D. Each device on the endpoint sends an RDI message to the device at the other endpoint, notifying it of the fault.

But, neither message gets through the fault between Node B and Node C. Neither message needs to. The devices at both ends of the link know the connection is broken because of the AIS messages they have already received. Manual loopback testing can begin from the devices at either end to determine the exact location of the fault.



A Fault on the Upstream Connection

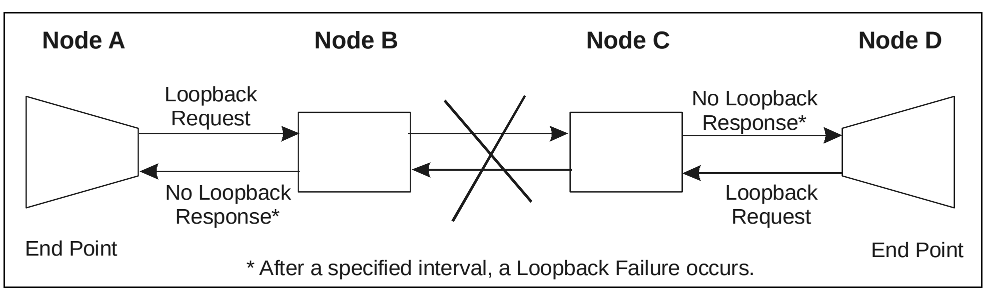
In the diagram above, the device at Node C senses the fault on the upstream between Node B and Node C. It sends an AIS message to the device on the endpoint at Node D. The device at Node D responds and sends a RDI message to the device on the other end of the WAN connection. The message goes through because no fault is present on the downstream part of the connection.

At this time, devices at both ends of the connection know the upstream path on the connection has a fault is broken, and two-traffic has stopped. Loopback testing can begin from the devices at either end to determine the exact location of the fault.

Transmitting and Receiving Loopback Messages

To determine the exact location of the fault, system administrators must use the OAM fault management function on one or both of the devices at the endpoints of the connection. Administrators can set a device to automatically send Requests for Loopback cells to a device at the other end of the connection on a regular basis; or they can set a device to send Requests for Loopback cells on a manual basis. Then administrators can read the counters for Requests for Loopback cells and the counters for Responses to Loopback Requests.

But to locate the fault, if the connection has multiple segments, someone must look at the counters for Requests and Responses on devices located on intervening segments. The fault on the connection will be between the last device to receive Requests for Loopback cells and the first device to not receive those same Requests for Loopback cells.



Lookback Request/No Response

Configuring the OAM Function

To configure the OAM function, first configure the ATM WAN port, then the WAN VC, and finally the OAM object itself.

1. Configure the ATM WAN port for AdminStatus using the object group, IfTable. If the status is not already Up, set it to Up.

   ```
   CRAFT> set cmiftable [1.2.1] adminstatus=up
   ```

2. Configure the ATM VC for AdminStatus, using the object group, cmAtmVcl. If the status is not already Up, set it to Up.

   ```
   CRAFT> set cmatmvcl [1.2.1.101] adminstatus=up
   ```

3. Configure the ATM WAN port for OAM AdminState, using the object group, cmAtmIfExt. If the status is not already Enabled, set it to Enabled.

   ```
   CRAFT> set cmatmifext [1.2.1] oamadminstate=enabled
   ```

4. Configure the ATM VCL for OAMAdminState, using the object group, cmAtmVcl. If the state is not already Enabled, set it to Enabled.

   ```
   CRAFT> set cmatmvcl [1.2.1.101]
   oamadminstate=enabled
   ```

5. Check your work in the cmAtmVcl group:

   ```
   CRAFT> get cmatmvcl [1.2.1.101]
   Group: cmAtmVclTable
   Instance: [1.2.1.101]
   PII                    =  1.2.1.101
   Vpi                    =  2
   Vci                    =  11
   AdminStatus            =  Up
   OperStatus             =  Up
   LastChange             = 12 day 1 hour 44 min 54.0 sec
                             (2000/09/28-14:44:46)
   AalType                =  Aal5
   Aal5CpcsTransmitSduS   =  1600
   Aal5CpcsReceiveSduSi   =  1600
   RowStatus              =  Active
   TransmitTrafficDescr   =  2
   OAMState               =  normal (0)
   OAMAdminState          =  enabled
   OAMAutoLBState         =  disabled
   OAMManualLBCmd         =  none
   OAMManualLBCmdStatus   =  none
   OAMLBInterval          =  17
   OAMLBTimeOut           =  18
   OAMTxAISCells          =  0
   OAMRxRDICells          =  0
   OAMRxAISCells          =  0
   OAMTxRDICells          =  0
   OAMTxLBRequestCells    =  0
   OAMRxLBResponseCells   =  0
   OAMRxLBRequestCells    =  0
   OAMTxLBResponseCells   =  0
   OAMRxUnsupportedCell   =  0
   OAMTxDiscards          =  0
   OAMRxDiscards          =  0
   ```

With this configuration, the OAM function will respond to AIS messages from the network by sending out RDI messages. It also will respond to Loopback Request messages sent by a device on the other end of the network. But, it will not send out Loopback Requests.

Configuring the OAM Loopback Function

The loopback function has two parts: an automatic function and a manual function. You can operate them at the same time or separately. The automatic function is simply for discovering when a fault occurs; the manual function is for discovering where the fault is on the connection.

Automatic Loopback Requests

The automatic loopback function will operate as long the port and the VC are UP and as long as the OAMAdminState function is Enabled. It will operate at the periods specified in the Interval and Timeout objects. It will record loopback failure in the OAM-State object.

To configure the loopback function within the OAM function, first configure the ports and VCs as described in the previous section, then the OAMAutoLBState object. You may want to make changes in the loopback interval and the loopback time-out. Follow these steps:

1. Check the ATM WAN port and the ATM VC for their AdminStatus. Make sure they are both set to Up.
2. Configure the ATM VC for OAMAutoLBState by setting it to Enable:

   ```
   CRAFT> set cmatmvcl [1.2.1.101]
   oamautolbstate=enabled
   ```
3. Check the OAMLBInterval object. Its default is 17 seconds; it has a range of 1 to 999 seconds. Reset it if necessary.

   OAMLBInterval specifies how often a Loopback Request will be sent until either a Loopback Response is received or a timeout occurs. If no response is received before timeout, the OAMState object indicates a loopback failure.
4. Check the OAMLBTimeOut object. Its default is 18; it has a range of 1 to 999 seconds. Reset it if necessary to a longer period.

   OAMLBTimeOut specifies how long Loopback Requests will be sent. The timeout period must always be longer than the interval period.

With this configuration, the OAM function will send out periodic f5End2End Loopback requests to the device on the far end of the network. If the device at the far end receives the requests—if no breaks exist on the connection—it will return Loopback responses during each period. If the device on the near end does not receive a response before the timeout period end, it records a loopback failure in the OAMState object.

Manual Loopback Requests

The manual loopback function will operate only when the port and the VC are UP, the OAMAdminState function is Enabled, and a command for the OAMManualLBCmd object has been sent. It will operate at the periods specified in the Interval and Timeout objects, that you either accepted or changed in the configuration of automatic loopback requests. The success or failure of a manually sent loopback will be recorded only in the OAMManualLBCmdStatus object, never in the OAMState object.

After the automatic function discovers the existence of a fault, this function will help you to discover which segment on the connection has the fault.

To send a manual loopback, follow these steps:

1. Set the OAMManualLBCmdStatus object to f5end2end:

   ```
   CRAFT> set cmatmvcl [1.2.1.101]
   oammanuallbcmd=f5end2end
   ```

2. Check the OAMManualLBCmdStatus object. You will see one of three conditions: in progress; succeeded; failed.

   ```
   CRAFT> get cmatmvcl [1.2.1.101]
   Group: cmAtmVclTable
   Instance: [1.2.1.101]
   PII                  =  1.2.1.101
   Vpi                  =  2
   Vci                  =  11
   AdminStatus          =  Up
   OperStatus           =  Up
   LastChange           =  12 day 1 hour 44 min 54.0 sec
                           (2000/09/28-14:44:46)
   AalType              =  Aal5
   Aal5CpcsTransmitSduS =  1600
   Aal5CpcsReceiveSduSi =  1600
   RowStatus            =  Active
   TransmitTrafficDescr =  2
   OAMState             =  normal (0)
   OAMAdminState        =  enabled
   OAMAutoLBState       =  enabled
   OAMManualLBCmd       =  f5end2end
   OAMManualLBCmdStatus =  succeeded
   OAMLBInterval        =  17
   OAMLBTimeOut         =  18
   OAMTxAISCells        =  0
   OAMRxRDICells        =  0
   OAMRxAISCells        =  0
   OAMTxRDICells        =  0
   OAMTxLBRequestCells  =  0
   OAMRxLBResponseCells =  0
   OAMRxLBRequestCells  =  0
   OAMTxLBResponseCells =  0
   OAMRxUnsupportedCell =  0
   OAMTxDiscards        =  0
   OAMRxDiscards        =  0
   ```

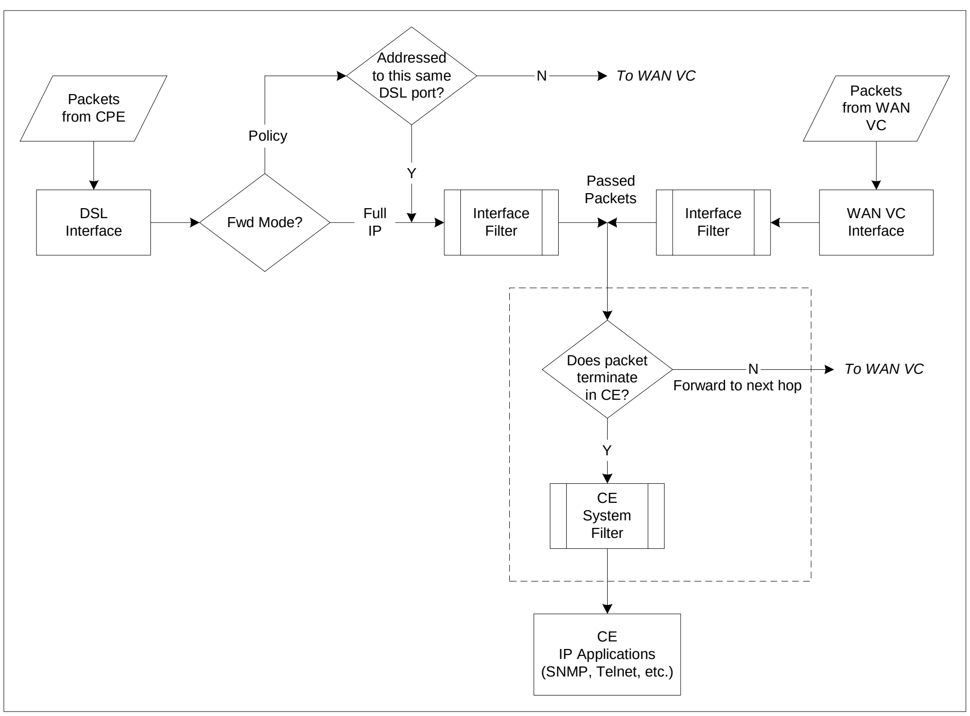
Interface Filters

When you configure a DSL interface with the IP netmodel, the CE150 allows all inbound messages to pass through without regard to IP address. Normally, the subscriber equipment processes only message packets that contain its IP address and ignores packets with a different address. Occasionally, data should not even *reach* a subscriber's equipment. A business, for example, would want to ensure that its message traffic is not accessible to its competition.

IP Filtering

IP filtering allows links to selectively pass or discard inbound IP packets based on parameters set by the operator. In general, filtering applies only to packets that are full-IP routed. The one exception is for packets that are policy routed to the IP address of the same DSL interface through which they entered the CE150 (such as a Ping from a device on a LAN at the customer premise to the IP address of its own associated CE150 DSL interface). Otherwise, policy-routed packets are not filtered.

Filtering cannot be done on interfaces with a networking model *other than* IP or HDIA. Also, filtering works only on *inbound* packets, before they reach the IP stack. Note, however, that packets may be inbound from either a WAN interface or from a DSL interface, provided they are destined for an IP address that terminates in the CE150 itself. The following flow chart shows the general flow of filtered packets in the CE150.



IP Filtering and Packet Flow in the CopperEdge DSLAM

Every source of IP packets can be identified by its PII. Each DSL port, network port, and every virtual circuit aggregated into a high-speed digital facility can be referenced by its PII. Each PII can be configured with a separate, independent set of IP filters.

Each filter, in turn, consists of a specific set of criteria against which incoming packets are compared. If the incoming packet matches the criteria specified by the filter, the packet is subjected to the action (Pass, Block, or Chain) specified by that filter.

When a packet arrives, it is checked against the filters in the list, in order. The first filter which matches the packet determines which action will be taken. An action code is also programmed into the filter, defining whether the matched packet will be passed or blocked, or if the filter will simply be combined with another (chained) to further specify the applicable range of packet values.

For maximum security, a packet which matches no filters is automatically blocked. However, *an interface for which no filters have been configured* ***passes*** *all packets.*

N O T E

Since IP packets can also be addressed to the CE150 itself, system security may be enhanced by establishing a list of filters applicable to the CE150. To configure filters that apply to the overall system, use the CE150's "virtual PII": 1.0.0.0. As a further security measure, and to counteract an inherent vulnerability in Internet-compliant systems, the cmFilter table for the CE150 includes a static filter that will immediately discard any ICMP Redirect messages destined for the CE150.

Viewing Filters

As we have seen, each interface can have its own set of filters. In certain cases, this list could be a long one. If you are using CopperView EMS to control the CE150, then you can display the entire contents of the filter table. Otherwise, you can retrieve them in batches using the Getall command:

```
CRAFT> getall cmfilter [1.4.7]
```

To view filters singly, begin by retrieving the first filter in the list:

```
CRAFT> get cmfilter [1.4.7, 1]
```

Then, to view succeeding filters for the same interface, simply use the Getnext command:

```
CRAFT> getnext cmfilter
```

Specifying, Activating, and Deleting Filters

Specifying the filters in the list, deleting them, or deleting an entire list, are accomplished using the cmFilter function object. For example:

```
CRAFT> set cmfilter [1.4.4, 3] function=delete
```

This command will delete filter number 3 for the interface specified (a DSL port at location 1.4.4). With the function object, the operator can command a filter or filter list to be deleted, or a filter to be inserted into a list. When the filter specifications are displayed in response to a Get or Getnext command, the filter-function always displays as `active`. The operator can set it to any value, to perform the following functions (recall that a filter definition must have both a PII and a filter-number):

- ACTIVE—Activates the filter criteria specified within the accompanying command (that is, in the same set statement as the active command). If there is already a filter at the location (list number) specified, it will be replaced with the filter you have defined. If you specify a filter-number larger than any on the existing list, the new filter will then be appended to the end of the filter-list.
- INSERT—Inserts the filter criteria specified within the accompanying command. The new filter will be assigned the filter number you specified and the numbers of all existing filters in the list will be increased by 1 to make room for the inserted filter. If you specify a filter-number larger than any on the list, the new filter will be added to the end of the filter-list.
- DELETE—Removes the specified filter. As shown in the example above, no other filter values should be entered. All filters following the deleted filter have their filter-numbers reduced by 1 as a result.
- DELETE_LIST—Deletes the entire filter list for the specified interface. No other filter values should be entered, but you must enter a filter-number (any 1- or 2-digit integer will do) because ‘filter-number’ is an index of the table.

Filter Criteria

Each IP filter includes the five filtering criteria listed below. In order for a packet to be considered as matching the filter, it must match *all five* of the categories. This means that all five items must be configured, even if the specification is *don't care*. For *don't care* fields, set the field to a value that will be true for every packet; for example, if you do not care about the Source Address, you can simply specify the source IP address and source subnet mask values as zero. Here are the five categories of filter criteria:

1. Source address and source mask.

   Match if (SA & mask) = value

2. Destination address and destination mask.

   Match if (DA & mask) = value

3. IP Protocol Identifier of the filter. The ID is entered as an integer which translates to the type of protocol as shown in the table below. A match occurs if:

   Filter Protocol ID = Protocol ID of the packet, or

   Filter Protocol ID = 0 (any).

   Valid IP Protocol IDs are shown in the following table:

Valid IP Protocol IDs

ID	Name	Description
0	IP	Pseudo protocol number; matches *any* Internet protocol
1	ICMP	Internet control message protocol
2	IGMP	Internet group multicast protocol
3	GGP	Gateway-gateway protocol
6	TCP	Transmission control protocol
17	UDP	User datagram protocol

4. Source port comparison.

   For TCP and UDP packets only, compares the source TCP/UDP port with the port specified by the filter. The comparison standard can be set to detect a match if the value of the packet's source port is:

   ANY anything
   LT less than the filter port
   GT greater than the filter port
   EQ equal to the filter port
   NE not equal to the filter port.

5. Destination port comparison.

   For TCP and UDP packets only, compares the destination TCP/UDP port of the packet with the port of the filter. The comparison standard can specify a match if the value of the packet's destination port is:

ANY	anything
LT	less than the filter port
GT	greater than the filter port
EQ	equal to the filter port
NE	not equal to the filter port.

N O T E

In specifying Source or Destination Port Comparisons, please be aware that the options "Any" and "NE" are homophones of each other. Always check for understanding to be sure that any oral instruction to configure this option as "Any" is not misinterpreted as "NE," or vice versa.

The *pass* and *block* actions are self-explanatory; the *chain* action, described below, is used to invoke multiple filtering criteria for the same packet, for example to specify a range of values.

Chaining Filters

The *Chain* action combines two or more sequential filters in a list. Only when a packet matches all filters in the chain will the action in the last filter in the chain be taken. Thus, to block packets with source ports between 120 and 130, you would combine two filters:

Match source_port > 119, action: CHAIN
Match source_port < 131, action: BLOCK

Specifying Filters

Remember, the first filter that matches the packet determines which action will be taken. So not only the selection criteria, but the order of their presentation is crucial to an effective filtering scheme.

To specify a complete packet filter, issue a Set command that includes all of the configurable objects in the cmFilter group, including the instance with its two index objects. For example:

```
*Index          = PII for this filter
*FilterNumber   = Filter Number for this filter
Function        = Sets status of the current filter or
                  filter list
```

```
SrcMask          = Source Subnet Mask Address
SrcAdrs          = Source IP Address
DstMask          = Destination Subnet Mask address
DstAdrs          = Destination IP Address
IpProtocol       = Filter Protocol Identifier
SrcPortCompare   = None, equal, less, greater, not equal
SrcPort          = Source port
DstPortCompare   = None, equal, less, greater, not equal
DstPort          = Destination Port
Action           = Filter Action: pass, block, chain
```

A full **set cmfilter** command would take the form shown in the following example:

```
CRAFT> set cmfilter [1.4.7, 1] function=insert
srcadrs=n.n.n.n srcmask=n.n.n.n dstadrs=n.n.n.n
dstmask=n.n.n.n ipprotocol=0 srcport=5 srcportcompare=eq
dstport=8 dstportcompare=ne action=pass
```

The cmFilter table is sorted by PII (SNMP first index), and then by filter-number (SNMP second index).

Sorting in the Filter Table

Index	filter_number	Filter Data
1.4.1	1	• • •
1.4.1	2	• • •
.4.1	3	• • •
1.4.2	1	• • •
1.4.2	2	• • •
:	:	:

N O T E

After any filter operation, that filter list is renumbered 1 through N, consecutively, with no gaps in the sequence. For example, if a filter list has five existing entries and you add filter number 99 to the list, it will become filter number 6. Use this method to add filters to the end of a list when you don't know exactly how many filters are on the list.

DSL IMUX

For subscriber endpoints equipped with the appropriate CPEs, the CE150 supports a technique called *Inverse Multiplexing*, or *IMUX*, in which multiple DSL ports are designated as members of a *bundle*, and treated as a single logical DSL port. The IMUX logical port and its DSL Multilink (multiple copper pairs originating at different physical DSL ports on the CE150, but terminating at a common endpoint CPE) combine to cumulatively increase both the bandwidth and the throughput of a single DSL link. Thus, an IMUX bundle containing two SDSL ports, each set at 1,568 Kbps, has a theoretical throughput of 3,136 Kbps. A small portion of the total, however, is reserved for management overhead. IMUX links may be either IDSL or SDSL, provided they are matched with the appropriate CPE and line module type.

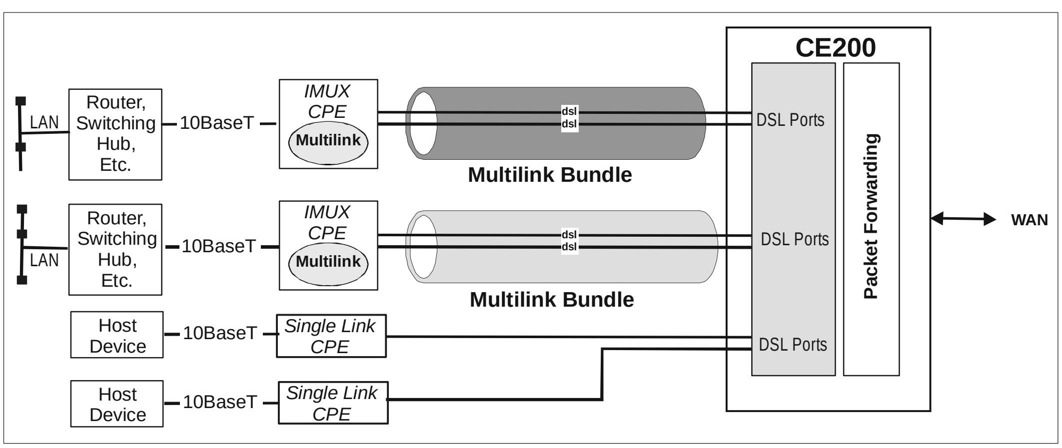
In the current release, the CE150 supports two DSL ports in one bundle. But the application is capable of supporting four-port IMUX bundles in software.

With few exceptions, IMUX bundles are treated in the same way as any other DSL port. Each bundled port is designated with its own PII and assigned to a logical slot 51, such as 1.51.3.

N O T E

Only IMUX-capable CPEs can be assigned as endpoints of an IMUX bundle; if a regular (single link) CPE is designated as the endpoint of an IMUX bundle and the CPE attempts to train, the connection will fail and the CE150 will generate an alarm. Moreover, any mismatch or conflict between the ports in an IMUX bundle and the bundle's endpoint CPE will result in an alarm condition.

In the following diagram, two-port DSL bundles are shown connecting the CE150 to separate IMUX CPEs. For contrast, two standard, single-port links are shown with their standard CPEs.



DSL Multilink (IMUX) Concept

Configuring an IMUX Bundle

In general, the procedure for setting up an IMUX bundle and configuring it is the same as for an individual DSL port, but there are a few special considerations.

When choosing which DSL ports to put into an IMUX bundle, note that they can be on the same DSL Module or on different ones, but all ports in the bundle must be on the same CE150.

1. Using cmHDSLModem, set the rates at which you want each of the ports in the bundle to operate.
2. For best multilink performance, set the two lines to the same speed. For example, if you want approximately 2000 Kbps throughput, it is better to set both lines at 1040 Kbps (for a total of 2080) than it is to set one at 1568 Kbps and the other at 416 Kbps (for a total of 1984).

   ```
   CRAFT> set cmhdsl [1.5.1] datarate=1040
   Set Successful

   CRAFT> set cmhdsl [1.5.2] datarate=1040
   Set Successful
   ```

3. Using cmBundle, create the bundle and enter the PIIs of the DSL ports on the CE150 that you want to assign to the bundle. You can enter all of the ports in the bundle with one command or you can enter one and the others later or you can create the bundle and add no ports initially.

   ```
   CRAFT> set cmbundle [1.51.1] member1pii=1.5.1
   member2pii=1.5.2
   Set Successful
   ```

4. If the CPE connected to the bundle is an IMUX CPE, then it will train normally, first on the Member1PII port, and then on the Member2PII port.
5. If the bundle is miswired or misconfigured, an alarm will be generated and a trap sent. If the configuration and wiring are correct, but there is no CPE connected or trained, the member ports in that bundle will show an endpoint ID consisting of zeros, and their status will indicate `WaitForAdd`. If a CPE is later connected, it will train normally and the member ports' endpoint and status information will automatically update.
6. Using cmIface, set the NetModel and the EncapsulationType for the IMUX bundle. The CopperRocket 202 works only with the Cross-Connect netmodel. Also, if you want to change the default for Service Class, you can do so at the same time. For example:

   ```
   CRAFT> set cmiface [1.51.1] netmodel=cross-connect
   encapsulationtype=q922
   Set Successful
   ```

N O T E

When creating IMUX bundles, do not attempt to preconfigure the individual DSL Member ports with cmIFace or cmSubIface before assigning them to a bundle. If you try to assign configured ports to an IMUX bundle, the cmBundle command will fail, and the system will return an error message. If this happens, use cmIface to reset the DSL member port causing the error to NetModel=None. Configuration of NetModel and the resultant forwarding mode can only be done by configuring the bundle.

7. Using cmSubIface, create the virtual circuit(s) for the IMUX bundle. In the process, set the destination PII to point to the desired virtual circuit on the WAN link.

```
CRAFT> set cmsubiface [1.51.1.16] rowstatus=create
destpii=1.2.1.200
Set Successful
```

8. Using frCircuit, create the entry for the virtual circuit on the WAN link if it does not already exist. At the same time, set the values for the Frame Relay congestion-management parameters.

```
CRAFT> set frcircuit [1.2.1.200] throughput=n1
committedburst=n2 excessburst=n3
Set Successful
```

DSL Voice and Data Service

In addition to conventional packet data traffic, the CE150 can accommodate voice over IP and other types of real time IP traffic, such as video conferencing. These real time packets and non-real time packets can coexist within the CE150, and both types of signals can share the composite data stream over DSL ports and WAN ports.

Both types of packets can also flow simultaneously over any of the CE150's Virtual Circuits: ATM, Frame Relay, or DSL. To optimize voice performance in this mixed environment, the CE150 provides for the assignment of various levels of priority to delay-sensitive, real time packets over non-real time packets.

Separate mechanisms are used for allocating priorities to traffic received over the DSL port (Class of Service), and traffic transmitted over the DSL port (Priority Queuing) without regard to the data rates on the various DSL links.

Class of Service

This function applies to packets *received* by the DSL ports on the CE150. It allows you to assign a level or class of service to each of the DSL ports, using *cmIface*. When you indicate the netmodel, the net mask, and the IP address for a DSL port, you also can enter a class of service for it. Typically, you will give a DSL port carrying voice a higher level of service than a DSL port carrying regular IP data traffic.

The CE150 allows you to set four levels of service: A, B, C, and D, where A is the highest class and D is the lowest. The default is D. The relative weight of each of the levels or classes is configured using the cmServiceClass MIB group. For more information on how these relative weights are derived, and how they affect their assigned interfaces, see the description in the cmServiceClass MIB group definition in the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual.

Once configured, the value selected as the weight for a particular class of service is the same for every DSL port to which it is applied. For example:

```
CRAFT> get cmiface [1.5.24]
Group: cmIfaceTable
Instance: [1.5.24.0]
PII                  = 1.5.24.0
IfIndex              = 1.5.24.0
Name                 = ....
GroupName            = ....
AdditionalInfo       = ....
NetModel             = IP
IpAddr               = 172.20.1.2
NetMask              = 255.255.255.0
MacAddr              = 0.60.58.1.0.76
BurnedInMacAddr      = 0.60.58.1.0.76
FarEndAddr           = 0.0.0.0
DestPii              = 1.2.1.43
CMCPCompatible       = Yes
```

```
EncapsulationType    = rfc1483
FwdMode              = IP-Policy
Pix                  = 3
Service Class        = A
```

In the example above, class of service on port 24 on the board occupying the slot 5 has been set to A, the highest class. To change the class of service on any of the other ports, use cmIface and change the defaults one at a time.

Class of service only comes into play when the CPU on the CE150 becomes congested. At that point, packets from ports with higher classes of service have a greater chance of being processed and passed on to the WAN interface. The higher the class of service—and the greater the weight of the higher class—the greater the chance of being processed.

Priority Queuing

Preference for delay-sensitive packets *transmitted* over the DSL port (from the CE150 to the subscriber interface) is provided through the Priority Queuing function. This configuration group applies only to DSL VCs that are created on the CE150, using the Cross-Connect netmodel. All of the other netmodels—HDIA, IP, VWAN, and CopperVPN—automatically give these kinds of packets priority in the downstream direction over DSL links. But, in the Cross-Connect netmodel, after you have set up the Permanent Virtual Circuits, you have to use cmSubIface to set each DSL VC for priority queuing: High for real time packets such as voice, low for non-real time packets.

The default is Low level. For a PVC that has voice packets or video conferencing packets and regular IP data packets on it, set the priority level to High. For a PVC with only regular IP data packets on it, leave the priority level at Low. For example:

```
CRAFT> get cmsubiface [1.4.24.16]
Group: cmSubIface
Instance: [1.4.24.16]
PII                  = 1.4.24.16
IfIndex              = 1.4.24.16
Name                 =
Priority             = High
RowStatus            = Active
```

In the example above, priority queuing on the 16th PVC on the 24th port on the board occupying the 8th slot has been set to High priority. To change the default on any of the other PVCs from the default (Low priority), you will have to use cmSubIface and change the defaults one at a time.

Priority queuing, once a PVC has been configured, comes into play whenever real time and non-real time packets are sent from the CE150 over the PVCs on the DSL port. Real Time packets will be processed before non-real time packets.

Radius Authentication

Enhanced system security and management of the CopperEdge user base is available through *Radius authentication*. All user and password databases on all CE150s on the network, including CopperView (SNMP) sessions, can be consolidated on a single remote Radius server. All login requests for CE150s then go to the Radius server for verification. A second Radius server supports a Radius accounting function. It keeps a record of user and SNMP access for each CE150.

N O T E

The CopperEdge does not send passwords for SNMP operators (such as Public and Private) in authentication requests.

Putting control at a regional or corporate level reduces chances of unauthorized entry and reduces the complexity of the user-management task. Up to three different server addresses can be specified for both the authentication server and the accounting server, enabling virtually 100 percent reliability.

N O T E

In the CopperEdge implementation of the Radius Accounting Function, the Radius Attribute Acct-Status-Type, used in the Radius accounting function, always has a value of 7 (Accounting On). Since the CE system does not utilize a shutdown process, the "Accounting Off" packet (Acct-Status-Type=8) is not supported. Instead, the server should be configured to treat Accounting On as signifying the beginning of a new session and the ending of the previous one.

Preparing to Configure CopperEdge for Radius Servers

Here is what you need to implement Radius in the CE150:

- You must have at least one Radius Authentication server; at least one accounting server is also required to implement the accounting functions
- The IP addresses of your Radius servers must be available through the CopperEdge routing table
- The Authentication Servers must be pre-configured to translate objects that are specific to the CopperEdge products, to object names recognized by the server
- The final step is to enable the cmRadius authentication object on the CopperEdge unit.

N O T E

Once the Radius server is provisioned and cmRadiusAuth is enabled, security features are under control of the Radius server. You can still configure the CopperCraft cmOperatorTable, but its contents will no longer be used. Operators configured in cmOperator and not contained in the Radius server database will not be able to log in. The Radius protocol requires that operator names and passwords are fully case-sensitive.

To make a request for authentication, the CopperEdge sends an Access-Request packet with the following data to the server:

Access-Request Packet Data

Attribute	Value	Note
User-Name	Text string input by user	Maximum of 32 characters.
User-Password	Text string input by user	Maximum of 32 characters. This field is one-way hashed using the MD5 algorithm. **Note:** The CE does not send passwords for SNMP operators (such as Public and Private) in authentication requests.
NAS-IP-Address	CE-IP-Address	This is the IP-Address of Ethernet port 1.2.1 or 1.15.1, depending on which SCM is being accessed by the operator. This field will contain 0.0.0.0 if there is no IP address configured for that port CE Management Ethernet IP address.
NAS-Identifier	CE-System-Name	This attribute is omitted if CE-System-Name is an empty string.

Upon receiving of the request by CopperEdge, the Radius server may respond with an Access_Accept packet with the following data in it:

Access-Accept Packet Data

Attribute	Value	Note
Idle-Timeout	Idle Timeout value in second (0 to 2147483647)	If this field is empty, the CE uses a predefined system Idle Timeout of 15 minutes.
Filter-Id	“Context = HH” **Note:** Be sure to enter this exactly as shown, with a space before and after the equal sign.	Where HH is a Hex number of the login context for this operator. Possible context is a bit map: 01 = CONTEXT_HTTP; 02 = CONTEXT_SNMP; 04 = CONTEXT_TELNET; 08 = CONTEXT_SERIAL; 10 = CONTEXT_SYSTEM; 20 = CONTEXT_FTP; 40 = CONTEXT_SHELL
Filter-Id	“Privilege = DD” **Note:** Be sure to enter this exactly as shown, with a space before and after the equal sign.	Where DD is a decimal number specified level of privilege designated to the particular operator. A number represents level of privilege: 1 = VIEW; 2 = MONITOR; 3 = PROVISION; 4 = SECURITY

Configuring Radius Servers for CopperEdge

Radius servers and their software are available in a number of different versions and implementations. Because setup and configuration procedures may vary, consult the documentation for whichever type you are using for full configuration particulars.

Similarly, while the CopperEdge Radius Client uses all the applicable Radius attributes as documented in RFC-2138 and RFC-2139, certain CopperEdge-specific parameters must first be translated so the server can interpret and deal with them appropriately:

1. Radius Attribute NAS-Identifier may be omitted in the packet if the CopperEdge SystemName object is empty.
2. The value of Radius Attribute *Filter-Id* substitutes for the CopperEdge Context and Privilege attributes (objects). Filter-Id attribute should contain the strings Context=HH (with HH a hexadecimal value) and Privilege=DD (with DD a decimal value) as described below. The syntax of the Filter-Id string is important because the CopperEdge unit must extract the values of HH and DD to properly complete the login process. Two Filter-Id strings are expected in an Access_Accept packet.

   The filter ID values for the CopperEdge Context attributes can be found on the previous page.

N O T E

Valid numeric values in the Context string can generally be combined to enable the same operator record to be used in multiple contexts. For example, decimal 12 (Hex 0c) enables both Telnet (04) and Serial (08) contexts. But to distinguish SNMP sessions from normal operator logins, and to prevent an operator from logging in using only the community string, SNMP user records must specify a Context of SNMP only. If an SNMP user has a login context value indicating anything other than SNMP, authentication will be rejected by the CopperEdge unit even if accepted by the RADIUS server.

The filter ID values for the CopperEdge Privilege attributes can be found on the previous page.

The Radius *Reply-Message* attribute may be configured with a text string (up to 128 characters) to serve as an error message in case an attempted login is denied (Access Reject). This string accompanies the Access_Reject packet, and is entered in the CopperEdge Event Log.

3. The Radius *Reply-Message* attribute may also contain a second 128-character (maximum) text string for use if you want the server to ask the user for additional information or to repeat the previous login attempt.

   The CopperEdge Radius client will respond to one Access_Challenge by resending the same UserName and Password. If the server does not accept it, the login fails. The string goes with the Access_Challenge packet. It appears in the CopperEdge Event Log.

4. Finally, the Radius server must use the same shared secret (AuthKey) string when configuring the both Primary and Secondary SCM clients. We recommend that you use the same AuthKey system-wide, but like a good password, it should be short-lived, unpredictable, and changed at irregular intervals.

Configuring CopperEdge for Radius Servers

When the Radius server is configured and ready to communicate, and its database has been provisioned with the user data base, the CopperEdge cmRadius MIB group can be configured:

1. With cmRadius in its default state (Authentication= Disabled), configure all of its objects except authentication. *Example:*

   ```
   CRAFT> set cmradius authkey=4cr37ei
   authprimaryipaddr=10.122.4.4 authprimaryport=1645
   acctprimaryipaddr=10.122.4.6 acctprimaryport=1646
   ```

   Be sure to use whichever port numbers are recognized by your Radius servers. Although the officially assigned port numbers for Radius Authentication and Accounting are 1812 and 1813, respectively, many current radius servers still use the port numbers of the original RFC: 1645 for authentication, and 1646 for accounting. Be sure that the numbers you assign in the CopperEdge cmRadius configuration match those used by your remote Radius servers.

2. When you are sure the rest of the configuration is complete and correct, activate the Radius capability:

   ```
   CRAFT> set cmradius authentication=enabled
   ```

For more data about the cmRadius MIB group, see the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual.

Saving Values

Here are the proprietary system values for the following configurable objects:

CalendarTime	Set or display the system real time clock (date and time set at initial configuration)
Redundancy	Enables or Disables redundancy functionality. When Enabled, the secondary Control and WAN complex can take over as primary. This is not available on the CE150.
Command	The command **get cmsystem** responds with the last operator command. Use the command **set cmsystem** to save the Config (configuration) file and to restart the system.

When you have finished adjusting the system attributes to your preferred settings, save the configuration data:

```
CRAFT> set cmsystem command=save
```

The data is saved to the appropriate volume (P:) in flash as: \ce200\system\config.tgz (or config.txt).

To verify that the save command was successful:

```
CRAFT> get cmsystem
Group: cmSystem
ObjectClass        =  System
OperState          =  Enabled
Version            =  E 3.0
Master             =  0.0.0.0
ConfigFileName     =  config.tgz
CalendarTime       =  2000/07/21-10:01:11
MyPII              =  1.2.0.0
PrimaryPII         =  1.2.0.0
SecondaryPII       =  0.0.0.0
Redundancy         =  NotAvailable
ShelfCount         =  1
ExpIpSubNet        =  192.168.250.0
ConfigSynch        =  Saved
Command            =  SaveConfig
CommandStatus      =  Succeeded
```

In the resulting cmSystem listing, the CommandStatus field should indicate `Succeeded`. If Command Status shows `In-Progress`, repeat the command **get cmsystem**; if Command Status shows `Failed`, contact Copper Mountain Tech Support.

N O T E

The cmSystem readConfig command is not supported in this release. If it becomes necessary to reload a previous configuration, use the cmMaintCmd group's ConfigRestore command. (See Restoring a Backed Up Configuration on page 111).

Restarting the System

The cmSystem group can also be used to do a warm restart of the system. A restart will interrupt any processes in progress, and the CE150 will reboot and reload its entire code base. Restart is *service-affecting*, with consequent (temporary) interruption of service for all subscribers connected to this CE150. Except in cases of emergency, restarts should only be attempted during scheduled maintenance periods at off-peak times.

To restart the system (using any of the available control interfaces):

```
CRAFT> set cmsystem command=restart
```

To restart the system if you have Security or System privileges:

```
CRAFT> scmrestart
```

Configuring Backup

Even with a modest number of connected subscribers, the saved file containing the CE150 configuration data can be extensive. Copper Mountain recommends that you periodically back up the file to a remote machine for reference (such as when the System Control Module is replaced) or emergency use.

With an on-line file server available to act as receiver, the CE150 (Version 2.0 and later) can be commanded to back up its current (saved) configuration on demand, or it can be configured to automatically send a Config file backup at specified recurring intervals. Using the *c*mMaintCmd table and its ConfigBackup command, you can specify the exact time to perform the backup, and how often, if ever, you wish it to recur.

N O T E

*If you back up or otherwise gain access to a config.tgz (or config.txt) file, you can view the file with a text editor, but do **not** try to edit it with any offline (non-CE150) application. While it may appear to be in text format, the config file cannot be reliably revised or updated with a text editor. If the file should become corrupted, routing information may be adversely affected, with unintended effects on service.*

Backing Up the Saved Config File

To configure a backup of the config.tgz (or config.txt) file, one or (preferably) two file servers (primary and secondary) need to be set up with FTP servers to receive the configuration file from the CE150.

The backed up file is always an uncompressed text file (config.txt).

Using the cmMaintCmd table and the configbackup command, configure the CE150 for config backup. Use the following command example as a model for your own configuration.

Full configuration of cmMaintCmd requires setting a fairly lengthy list of parameters. To reduce the required complexity of any single command, you can issue multiple Set commands, provided you begin by specifying the ConfigBackup command and creating the appropriate row in the cmMaintCmd table.

```
CRAFT> set cmmaintcmd [configbackup] rowstatus=createandwait

CRAFT> set cmmaintcmd [configbackup]
primaryipaddr=206.71.190.4

CRAFT> set cmmaintcmd [configbackup]
secondaryipaddr=206.71.190.5

CRAFT> set cmmaintcmd [configbackup] directory= /
ce200config/sys1
```

```
CRAFT> set cmmaintcmd [configbackup] basefilename=sysdeltabu

CRAFT> set cmmaintcmd [configbackup] username=anonymous
pass=""

CRAFT> set cmmaintcmd [configbackup] recurrence=168

CRAFT> set cmmaintcmd [configbackup] start=99/04/01-03:20

CRAFT> set cmmaintcmd [configbackup] rowstatus=active
```

In the above example, the CE150 has both primary and secondary file servers for its FTP transaction, which is set to recur weekly (recurrence=168 hours) at 0320 hours (3:20 a.m.).

If you set the recurrence at 0 and the start time at 0 or any time in the past, the command will run immediately upon activation of the row.

In a recurring configuration backup, you would specify the time of the first occurrence, and an interval for subsequent file uploads. When configured, the CE150 will send the current configuration saved in its flash. *A recurring backup will only be attempted if this is the first attempt since powerup, or if the config file saved in flash has changed since the previous backup.*

If a fresh backup is needed, the CE150 attempts to transfer the config to the primary file server. If the transfer succeeds, the CE150 logs an informational event, and sends an SNMP trap. If the transfer to the primary fails, the CE150 will also log and trap the event, and will then attempt to save the config to the secondary server. If the transfer to the secondary server fails, the CE150 will log the event as a minor alarm and send the applicable trap.

For more information about the maintenance commands, see the description of cmMaintCmd in the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual.

Restoring a Backed Up Configuration

Restoring a configuration is very similar to backing it up, except that the configuration data is flowing back toward the CE150 rather than toward the file server. The ConfigRestore command is also slightly simpler, as there is no need to specify a secondary server or recurrence-related objects.

As is the case with the configbackup command, you can issue multiple Set commands to set up the appropriate row in the cmMaintCmd table.

```
CRAFT> set cmmaintcmd [configrestore] rowstatus=
notinservice

CRAFT> set cmmaintcmd [configrestore] primaryipaddr=
206.71.190.4

CRAFT> set cmmaintcmd [configrestore] directory=
/ce200config/sys1
```

```
CRAFT> set cmmaintcmd [configrestore] basefilename=
config206_71_190_419991101_0000

CRAFT> set cmmaintcmd [configrestore] username= anonymous
password=”“

CRAFT> set cmmaintcmd [configrestore] rowstatus= active
```

The final command in the string will activate the config restore, and the file will be downloaded to the CE150 flash, but will not be loaded into the active system. To complete the reload, you must reset the system:

```
CRAFT> set cmsystem command=restart
```

The restored configuration file is always a compressed binary file.

Chapter 5
Routing, Forwarding, Managing Links

This chapter discusses the 200CE150's network interfaces and considerations in routing DSL links. It describes ATM and Frame Relay functionality as implemented in the CE150, and provides instructions with examples for configuring circuits and connections for communication over Frame Relay and ATM links, and over the Ethernet when that interface is used for network access.

A reference listing of all of the CE150 MIB objects, including those related to ATM and Frame Relay functionality (configuration, status, and performance monitoring) is contained in the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual.

Features

No special commands or objects are needed for configuring either Frame Relay or ATM WAN VCs. All parameters are accessible from the existing CE150 MIB tables.

DS3 Frame Relay Module

The Frame Relay WAN VCs currently support:

- RFC-1973, point-to-point protocol over Frame Relay
- RFC-1483, encapsulation (FUNI)
- RFC-1490, bridged or routed-IP encapsulation
- NNI on WAN Links
- RFC-1315, DLCMI and link (PVC) performance/error monitoring
- Traffic Shaping, with settings for Committed Information Rate (CIR), Committed Bursts (CB), and Excess Bursts (EB)
- Link Management Interface (LMI), with protocols LMI Rev1, ANSIT1-617-D, and Q9.33-Annex-A
- IP packet filtering on VCs
-

DS3 ATM Module

The ATM WAN VCs currently support:

- RFC-1483, multi-protocol encapsulation over ATM
- RFC-1490, multi-protocol encapsulation over Frame
- RFC-2364, point-to-point protocol over ATM
- ATM Permanent Virtual Circuits (Switched Virtual Circuits not supported at this time)
- ATE, functioning as an ATM Host
- Traffic Shaping, with service categories (ubr, nrtVBR, rtVBR) and with settings for Peak Cell Rate (PCR), Sustained Cell Rate (SCR), Maximum Burst Size (MBS), and CDV (Cell Delay Variation)
- OAM Fault Management, with Alarm Indication Signals (AISs) and Remote Defect Indications (RDIs) along with functions for automatic and manual (diagnostic) loopbacks
- virtual connections (no virtual path tunneling
- IP packet filtering on VCs
-

DS1 Frame Relay Module (or Quad T1)

The ATM WAN VCs currently support:

- RFC-1973, point-to-point protocol over Frame Relay
- RFC-1483, encapsulation (FUNI)
- RFC-1490, bridged or routed-IP encapsulation
- NNI on WAN Links
- RFC-1315, DLCMI and link (PVC) performance/error monitoring
- Traffic Shaping, with settings for Committed Information Rate (CIR), Committed Bursts (CB), and Excess Bursts (EB)
- Link Management Interface (LMI), with protocols including LMI Rev1, ANSIT1-617-D, and Q9.33-Annex-A
- IP packet filtering on VCs identical to that on DSL links
- Receive timing from a clock recovered from the network line signal; transmit timing from a local/internal clock or a transmit clock recovered from the network.
- ESF framing mode
- Loopbacks, with settings for inward loop, near-end line loop, near-end payload loop, far-end line loop, and far-end payload loop

Configuring CE150 Network Interfaces

You can review the basic configuration of any CE150 interface by checking the contents of its entry (PII) in the *c*mIface table (see the sample table entry below). Then, use the command, **`set cmiface`**, to specify the interface and configure its various parameters (objects).

In the following example (and in the tables throughout this chapter), shaded objects are read-only; you cannot directly set them. The objects without shading are operator-configurable.

```
Group: cmIfaceTable
Instance: [1.4.1.0]
PII                        =  1.4.1.0
IfIndex                    =  1.4.1.0
Name                       =  Name of this port
AdditionalInfo             =  ....
NetModel                   =  IP
IpAddr                     =  192.168.1.1
NetMask                    =  255.255.255.0
MacAddr                    =  0.60.58.0.0.b
BurnedInMacAddr            =  0.60.58.0.0.b
FarEndAddr                 =  0.0.0.0
DestPii                    =  1.2.2.32
CMCPCompatible             =  Yes
EncapsulationType          =  rfc1483
FwdMode                    =  IP Policy
Pix                        =  54
```

Full IP vs. Policy-Routed Links

As described in the previous chapter, you can configure a DSL link to make the CPE behave as an ordinary IP router. If you configure a DSL link with an IP address, and you also specify a destination PII. At that point, the link is said to be *IP-Policy routed* and the status of the fwdMode field in the cmIface table will display `IP Policy`.

Policy routing is a forwarding process that ignores the destination IP address in packets. Instead, packets are sent to the destination PII specified by the configuration of the packet's source PII (the DSL interface). If no destPII is specified, the link is Full-IP routed.

Policy routing allows you to enhance security between DSL links, since traffic arriving on one DSL link cannot be routed out to another DSL link nor to any other IP address, no matter what its contents. The one exception to the Policy Routing scheme occurs when a premise host sends a packet to the CE150's IP address. In that case, the host is talking *to* the CE150 rather than *through* it.

The CE150 receives and processes the packet, but it is not policy routed. Such a setup allows a host on a LAN served by a CPE to ping the DSL interface.

N O T E

In the IP networking model, packets from the WAN interface or any non-DSL interface are always full-IP routed; i.e., the system sends the packet to a DSL link (or elsewhere) based on its destination IP address. Only packets arriving on DSL links can be policy routed.

Configuring Full-IP DSL Links

You configure DSL links for Full-IP routing as described in the previous chapter. In brief, assign the DSL interface an IP address, optional subnet mask, and leave the DestPii object set to zero. (FarEndAddr is normally ignored on DSL links, unless configured for policy routing over Ethernet). Be sure to configure the EncapsulationType, NetModel, and any other mandatory settings, as well. For example:

```
Group: cmIfaceTable
Instance: [1.4.1.0]
PII                      =  1.4.1.0
IfIndex                  =  1.4.1.0
Name                     =  System-assigned port name
GroupName                =  User-Group ID String
AdditionalInfo           =  User-defined text string
NetModel                 =  IP
IpAddr                   =  192.168.1.1
NetMask                  =  255.255.255.0
MacAddr                  =  0.60.58.0.0.b
BurnedInMacAddr          =  0.60.58.0.0.b
FarEndAddr               =  0.0.0.0
DestPii                  =  0.0.0.0
CMCPCompatible           =  Yes
EncapsulationType        =  rfc1483
FwdMode                  =  Full-IP
Pix                      =  88
```

You can change a policy-routed DSL link to full-IP at any time by setting destPii to **`0.0.0.0`**. The fwdMode field will display the mode (`Full-IP` or `IP Policy`) based on the setting of the destPii object.

Configuring Policy-Routed DSL Links

DSL links may be policy routed to a virtual circuit on a CE150 WAN port, or to a specific host device on a LAN over the CE150 Ethernet port.

IP Policy Over WAN

To configure a DSL link for policy routing over a WAN link, configure the DSL link (**`set cmiface`**) to specify its IP address, net mask, and a DestPii. The DestPii includes shelf number, slot, port and the virtual identifier (Frame Relay DLCI or ATM virtual circuit number), in the full PII format as shown in the following sample configuration. For example:

```
Group: cmIfaceTable
Instance: [1.4.1.0]
PII                      =  1.4.1.0
IfIndex                  =  1.4.1.0
NetModel                 =  IP
IpAddr                   =  192.168.1.1
NetMask                  =  255.255.255.0
MacAddr                  =  0.60.58.0.0.b
BurnedInMacAddr          =  0.60.58.0.0.b
FarEndAddr               =  0.0.0.0
DestPii                  =  1.2.1.32
CMCPCompatible           =  Yes
EncapsulationType        =  rfc1483
FwdMode                  =  IP Policy
Pix                      =  27
```

As shown in this example, policy-routed DSL links *do* have IP addresses and subnet masks, because packets are routed *to* them with full-IP routing.

Packets forwarded in IP-Policy mode are capable of generating the IETF standard TTL Exceeded ICMP error message if the packets expire before reaching their destination (V1.6 and later). This capability enables use of standard TraceRoute diagnostic utilities.

Note, however, that because outbound policy-routed packets bypass the CE150 routing table, a TraceRoute sequence cannot identify the CE150 leg of the route. To resolve the hop to and from the CE150, you can add an entry to the routing table that will let the ICMP error message track through and identify the route segment from the WAN VC, through the CE150, to the DSL port/CPE.

IP Policy Over Ethernet

N O T E

Performance implications when using the Ethernet port of the CE150 as a WAN interface, using the IP Policy netmodel, are not clear. If you establish other network connections over a WAN port (FR, ATM, V.35) along with a connection over the Ethernet port, you may have performance degradation on one link or all of the links. Also, restriction to a single virtual bridge group in VWAN over Ethernet may present a security problem on your network.

Any DSL link can also be policy routed to a specific host on an Ethernet LAN through the CE150 Network (Ethernet) port as follows.

First, if the CE150 Ethernet port has not yet been configured, assign its IP addres. Second, set its networking model and encapsulation type parameters. For example:

```
CRAFT> set cmiface [1.3.1] ipaddr=192.168.1.1
netmask=255.255.255.0 netmodel=ip encapsulationtype=none
Set Successful

CRAFT> get cmiface [1.3.1]
```

The response will display all objects in the cmIface group for the specific instance (the Ethernet interface).

```
Group: cmIfaceTable
Instance: [1.3.1]
PII                     =  1.3.1.0
IfIndex                 =  1.3.1.0
Name                    =  EthernetPort
AdditionalInfo          =  System Master
NetModel                =  IP
IpAddr                  =  192.168.1.1
NetMask                 =  255.255.255.0
MacAddr                 =  0.80.56.34.da.87
BurnedInMacAddr         =  0.80.56.34.da.87
FarEndAddr              =  0.0.0.0
DestPii                 =  0.0.0.0
CMCPCompatible          =  No
EncapsulationType       =  None
FwdMode                 =  Full-IP
Pix                     =  1
```

As shown, the CE150 Ethernet port is configured as full IP. To configure the DSL port for policy routing, configure its IP address and netmask, destination PII (the System Control Module Ethernet port, 1.2.1) and far-end address (the IP address of the Ethernet host machine). For example:

```
CRAFT> set cmiface [1.4.1] ipaddr=200.10.10.1
netmask=255.255.255.252 destpii=1.3.1 farendaddr=192.168.1.2
netmodel=ip encapsulationtype=rfc1483
Set Successful
```

Note that this use of FarEndAddr is very different from its function in configuring WAN virtual circuits. Note also that, if the DestPII is the Ethernet port on the System Control Module, then the FarEndAddr must be a separate IP address, but it must be on the same subnet as the Ethernet Port.

The configured DSL interface should be similar to the following example.

```
Group: cmIfaceTable
Instance: [1.4.1]
PII                       =  1.4.1
IfIndex                   =  1.4.1
NetModel                  =  IP
IpAddr                    =  200.10.10.1
NetMask                   =  255.255.255.252
MacAddr                   =  0.60.12.34.56.78
BurnedInMacAddr           =  0.60.12.34.56.78
FarEndAddr                =  192.168.1.2
DestPii                   =  1.3.1.0
CMCPCompatible            =  Yes
EncapsulationType         =  rfc1483
FwdMode                   =  Policy-IP
Pix                       =  14
```

A key difference between policy routing over Ethernet versus policy routing over WAN is that over Ethernet, the far-end address of the DSL interface is configured with the IP address of the Ethernet host. In contrast, over a WAN link, the far-end address (FarEndAddr) is not significant for the DSL link. You don’t need to configure it.

Configuring Ports for VWAN Over Ethernet

Configuration of DSL ports and WAN VCs for VWAN is described in Chapter 4. Although the Ethernet port does not support virtual circuits, it must still be configured in order to function properly in the VWAN networking model.

If the CE150 Ethernet port has not yet been configured, explicitly assign its networking model and encapsulation type. If the Ethernet port will be used for system management, you will also need to specify its IP address and mask. For example:

```
CRAFT> set cmiface [1.3.1] ipaddr=192.168.1.1
netmask=255.255.255.253 netmodel=vwan encapsulationtype=none
Set Successful

CRAFT> get cmiface [1.3.1]
```

The response will display all objects in the cmIface group for the specified instance (the Ethernet interface).

```
Group: cmIfaceTable
Instance: [1.3.1.0]
PII                      =  1.3.1.0
IfIndex                  =  1.3.1.0
Name                     =  EthernetPort
GroupName                =  Garfield
AdditionalInfo           =  Brighton AP Office
NetModel                 =  VWAN
IpAddr                   =  192.168.1.1
NetMask                  =  255.255.255.0
MacAddr                  =  0.80.56.34.da.87
BurnedInMacAddr          =  0.80.56.34.da.87
FarEndAddr               =  0.0.0.0
DestPii                  =  0.0.0.0
CMCPCompatible           =  No
EncapsulationType        =  None
FwdMode                  =  VWAN bridge
Pix                      =  1
ServiceClass             =  D
```

VWAN over Ethernet functions as a “nailed-up” point-to-point link to a single router, and there can be no other device on the LAN. Note that the Ethernet port, when configured for VWAN, operates in promiscuous mode. Thus, *you cannot specify an IP address for the router*. Rather the CE150 learns the MAC address of the router from the received packet stream.

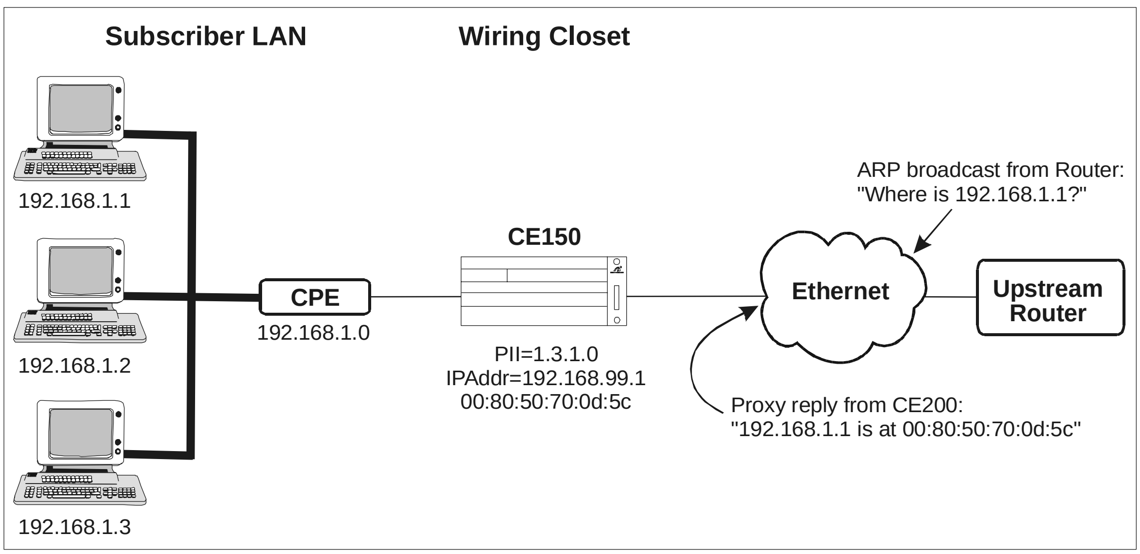
Note also, that the forwarding mode over Ethernet is always displayed as `bridge`, regardless of the number of DSL ports served.

Proxy ARP

For DSL interfaces that are routed over the 10/100BASE-T Ethernet interface, the CE150 supports proxy ARP functionality. The feature simplifies configuration by allowing the CE150, when queried, to use its own Ethernet (MAC) address as a proxy for LAN destinations on its connected CPE.

Proxy ARP for Ethernet is actually the reverse of ARP used in the CopperVPN networking model and described in Chapter 4. In CopperVPN, the CE150 replies to queries from CPEs on LANs as a proxy for the upstream router. In Proxy ARP for Ethernet, the CE150 responds to queries from the upstream device as a proxy for CPEs on LANs.

The diagram below illustrates Proxy ARP for Ethernet. But Proxy ARP is valid only if the DSL port is configured with NetModel=IP and also if the network or WAN connection is through the 10/100BASE-T Ethernet interface on the primary System Control Module. For details on configuring DSL ports for Proxy ARP, see the cmProxyARPTable in the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual.



CE150 Proxy ARP

Configuring WAN VCs

Virtual Circuits, either Frame Relay or ATM, used only as destinations of policy IP routes do not need to be elaborately configured. For those VCs, simply specify the Frame Relay Data Link Circuit Identifier (DLCI) or ATM virtual circuit number associated with the circuit in the destPii of the DSL links. Activate the circuit as described below.

In this mode, the CE150 has no knowledge of IP addresses at the far end of the VC. As a result, the CE150 is fully independent of the far-end IP numbering scheme. In fact, two different VCs can have duplicate, overlapping IP addresses (impossible with full IP routing), and the CE150 will not be affected.

If you wish, you can give IP addresses to the VCs. The CE150 treats each VC as a separate point-to-point (unnumbered) IP link. Unnumbered IP links are ones with no IP subnet number. Each end of an unnumbered IP link may have an IP address, and if both are present, the two addresses can be on different subnets.

However, the preferred way to configure an unnumbered IP link is to assign it a FarEndAddr, but not an IP address. The FarEndAddr tells the CE150 what the IP address of the device at the far end of the point-to-point link is. The WAN port at the near (CE150) end of the link typically has no use for an IP address. See the configuration below:

```
Group: cmIfaceTable
Instance: [1.2.1.32]
PII                       =  1.2.1.32
IfIndex                   =  1.2.1.32
Name                      =  ....
AdditionalInfo            =  ....
GroupName                 =  ....
NetModel                  =  IP
IpAddr                    =  0.0.0.0
NetMask                   =  0.0.0.0
MacAddr                   =  ff.ff.ff.ff.ff.ff
BurnedInMacAddr           =  ff.ff.ff.ff.ff.ff
FarEndAddr                =  192.166.101.8
DestPii                   =  0.0.0.0
CMCPCompatible            =  No
EncapsulationType         =  rfc1490
FwdMode                   =  Full-IP
Pix                       =  141
```

Once the FarEndAddr is defined for the VCs, you can use those IP addresses as the next-hop for route table entries. For more information, see the ipRoute table description in the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual. For example:

```
CRAFT> set iproute [192.168.250.0] nexthop=192.166.101.8
```

Currently, when you delete a FarEndAddr (by setting it to 0.0.0.0), you must manually delete any associated routes in the ipRoute table. For example, if you move an IP address from one VC to another, you must first delete, and then recreate any routes using that address. You can delete routes at any time, but you cannot create a new route until *after* the interface IP address is assigned.

N O T E

You can make a VC both a policy-route destination and a full-IP interface. The two functions are independent of each other.

Configuring DS3 Frame Relay

To establish a Frame Relay virtual circuit on a WAN interface, first, *enable* the WAN port for Frame Relay. Second, specify the LMI scheme, if any. VCs now can be specified and activated.

Initial Configuration

To create a Frame Relay link, you must use two similarly named MIB groups: frDlcmi (from the Frame Relay MIB) and cmFrDlcmi (the Copper Mountain MIB). Both groups configure links on the physical *port*, and thus apply to all of the Frame Relay virtual circuits that you may subsequently assign to that port. For example, to enable a link on a WAN port with no LMI:

```
CRAFT> set cmfrdlcmi [1.2.1] adminstate=enable
```

To provide a link with LMI, use the State object in the standard frDlcmi group to specify the LMI formatting scheme. Available options are: ansiT1-617-D (ansiT1.617 Annex D), lmirev1 (LMI Revision 1), Q9.33-Annex-A, and noLmiConfigured. For example:

```
CRAFT> set frdlcmi [1.2.1] state=ansit1-617-d
```

N O T E

Always disable the link using cmfrdlcmi (Admin=dis) before changing the LMI state or any other link attributes. Modifying link attributes is service-affecting.

When you enable LMI on a WAN port, remember to configure your Frame Relay router as DCE LMI.

Adding a PVC

Once you have enabled a link as described above, individual WAN PVCs can be configured. Do this using the frCircuit group to specify and activate the circuit numbers. For Frame Relay VCs, you can use any DLCI from 16 to 991.

To establish a VC (DLCI 32, port 1 in the example):

```
CRAFT> set frcircuit [1.2.1.32] state=active
```

Note that the index in this case includes the complete PII entry, that is the interface location (1.3.1) and the PVC number (32).

To disable PVC 32 on port 1:

```
CRAFT> set frcircuit [1.2.1.32] state=inactive
```

To delete PVC 32:

```
CRAFT> set frcircuit [1.2.1.32] state=invalid
```

N O T E

Before a PVC can be deleted, its link must be disabled through cmFrDlcmi. An active circuit on an enabled link cannot be deleted.

Configure the Frame Relay DCE to Connect with the CE150

The CE150 behaves as a Frame Relay UNI DTE (User Network Interface, Data Terminal Equipment). Therefore, the CE150 must connect to a Frame Relay UNI DCE (Data Communications Equipment) interface, such as a Frame Relay switch. The DCE provides clocking for the CE150 WAN interface.

Throughput Management

For any Frame Relay link (PVC), only a finite amount of data can be accommodated between any two points in time. If the network becomes congested, connected devices at one or both ends of a PVC are automatically notified by the network that the volume of transmitted data will be temporarily retarded until the congestion is alleviated. To ensure a continuous and orderly flow of data, even during these periods of congestion, Frame Relay relies on the concept of a *Committed Information Rate* (*CIR*).

The CE150 supports three configurable command objects related to Committed Information Rate, all through the frCircuit group used to configure Frame Relay virtual circuits (PVCs) established at *WAN* interfaces. They are:

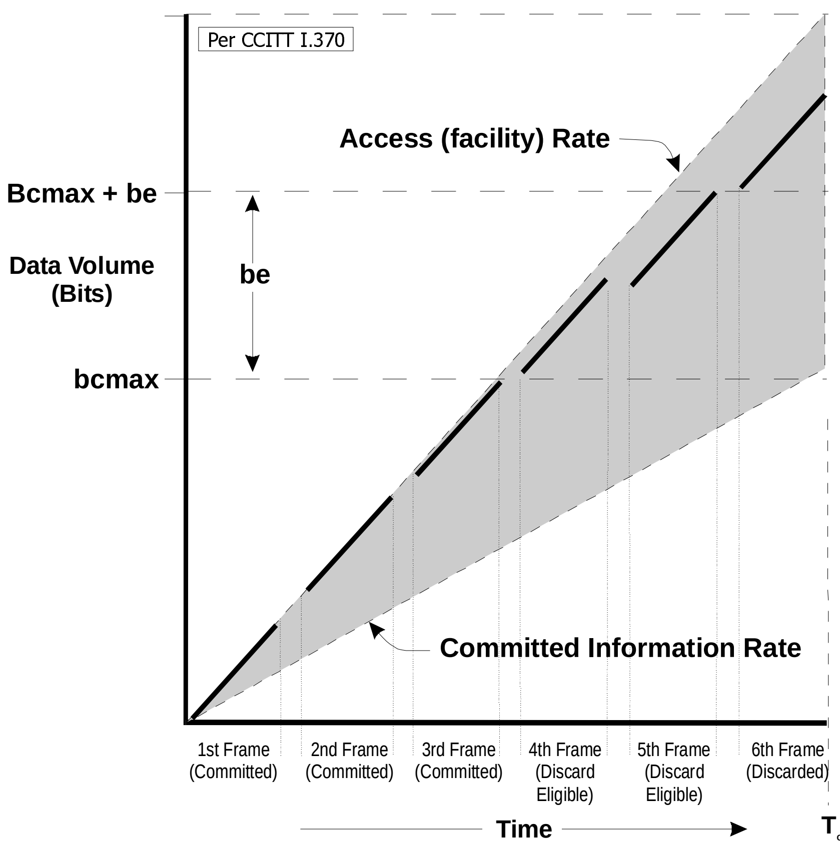
- Throughput
- CommittedBurst
- ExcessBurst

Throughput

Throughput (a term synonymous with CIR) specifies the average amount of data, in bits-per-second, that can be sent over a given PVC (assuming a reasonably well designed network) on demand, 24 hours a day. Since PVCs may be shared by a number of users, the Committed Information Rate for a CE150 PVC will typically approximate some reasonable sum of the data rates of all of the subscribers who have access to the interface.

As a practical matter, however, a user of a real-world data link can almost always transmit and receive at a much faster rate than the nominal baseline indicated by the CIR. As we know, most networked data communication is bursty in nature, that is, it consists of bursts of packets (some of which may be very large), separated by relatively long idle periods. Thus, there is another, typically larger, number that represents a burst of data which the network will accept with a reasonable probability of reliable delivery. This is the CE150 *CommittedBurst (bcMax*).

The third congestion-management setting specifies the amount of data which, though in excess of the Committed Burst, can still be handled by the network and the circuit under normal conditions. This amount is the *ExcessBurst* (*BE)*. However, because this ExcessBurst does exceed the commitment, it necessarily is at greater risk of causing or encountering congestion on the link. Frames that fall within the BE range are automatically marked with the *discard-eligible* (*DE*) bit, which means that if the network becomes congested, the frame will simply be dropped. Data within the commitment time interval that exceed the value of BE will be discarded, regardless of whether the link is congested or not. The following illustration, adopted from the CCITT standard on congestion management, shows the relationship between the three variables within the commitment interval (T_c).



Throughput Management Variables

A definition for the commitment interval has been established by CCITT and adopted by ANSI. The actual value will depend on the settings of *CIR*, *bcMax* and *be*. For details on calculating T_c, refer to the Throughput object in the frCircuitTable in the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual.

Note that while the values of bcMax and BE remain constant over time, the values of CIR and Access Rate appear as slopes. This is because bcMax and BE are expressed as absolute values (fixed number of bits transmitted over the commitment-interval period), while CIR and Access Rate are expressed in bits per second.

Setting CIR Parameters for Frame Relay PVCs

By default, the CIR setting on newly created WAN PVCs is zero. The default value of bcMax is also zero. In this case, ExcessBurst automatically assumes the value of the Speed object in the IfTable. Thus, Speed in this instance may be considered as synonymous with Access Rate as shown in the preceding illustration.

N O T E

Congestion management parameters (CIR, bcMax, and BE) are not configurable for DSL VCs. Since there is no enforcement of CIR on the receive side of WAN VCs, bandwidth commitments cannot be assigned or honored on the transmit side of the corresponding DSL VCs.

With the factory default settings, every frame is technically discard-eligible. But, as frames will not normally be discarded unless the Access Rate is exceeded, the default scenario represents the most conservative situation.

If you decide to configure the congestion-management parameters to a value other than the default, then we recommend that you first agree upon appropriate values for CIR, bcMax and BE with your network provider.

To set the information commitment parameters, all three of the relevant objects *must* be set as part of the same command to ensure that all three are simultaneously applied. For example:

```
CRAFT> set frcircuit [1.2.1.101] throughput=n1
committedburst=n2 excessburst=n3
```

Where:

n1 is the Throughput value in bits per second
n2 is the size of the maximum committed burst in bits
n3 is the size of the ExcessBurst credit in bits

For more information about throughput management and the tools available to monitor congestion on your Frame Relay links, see frCircuitTable in the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual.

Configuring DS3 ATM

The CE150 provides concentration of DSL links originating at the Customer Premises onto DS3 ATM trunks. If you have configured ATM for other types of equipment, you will likely notice that the ATM MIB used here comprises only a small subset of the full MIB. Since the CE150 is an ATM host and not a switch, configuration is relatively simple. In fact, only the atmInterfaceConfTable was extracted from the ATM MIB. Two tables were also added to the proprietary cm MIB, cmAtmVclTable and cmDS3Atm, to support the CE150's implementation of ATM.

The current release will support up to 976 ATM VCs and up to 16 Virtual Paths per DS3 ATM module. The CE150 software supports RFC-1483 encapsulated bridge and IP traffic, PPP over ATM, and an implementation of FRF.8 and FRF.5.

The DS3 ATM Module has the same loopback options (LineLoop, InwardLoop, NoLoopback) and performance parameters (through the dsx3 MIB objects) as the DS3 Frame Module, except that dsx3 table listings for PESs, PSESs, and PCVs will always be zero, as those objects apply only when the LineType is M23. The M23 line type is not supported by the CE150's DS3 ATM Modules.

Configuring ATM VCs

To configure a WAN link for ATM, the appropriate DS3 ATM WAN module must be installed and operating. The DS3 link must also be set to use your selected timing source (local or loop). For example:

```
CRAFT> set dsx3config [1.2.1] transmitclocksource=loop
```

Next, use cmDS3ATM to configure scrambling and cell mapping. For example:

```
CRAFT> set cmds3atm [1.2.1] cellpayloadscrambling=enabled
cellmapping=plcp
```

Configuration of the ATM interface and its circuits can then proceed as follows:

1. Configure the physical interface. For example:

   ```
   CRAFT> set cmiface [1.2.1] netmodel=none
   encapsulationtype=rfc1483
   ```

2. Create and configure the ATM VCL. For example:

   ```
   CRAFT> set cmatmvcl [1.2.1.66] rowstatus=create vpi=0
   vci=55 adminstatus=up
   ```

3. Configure the protocol interface (*cmIface*) for the ATM VC. For example:

   ```
   CRAFT> set cmiface [1.2.1.66] netmodel=ip
   farendaddr=192.166.100.1 encapsulationtype=rfc1483
   ```

4. The above example configures an ATM VC for the IP netmodel.

NOTE

If the DS3 ATM physical port is disabled, the port does not go idle, but transmits AIS signals.

For more information about the ATM-related objects in these examples and their configuration options, see the MIB Definitions in the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual.

Configuring ATM VCs for Quality of Service

The CE150 supports ATM quality of service (QoS) for all cells transmitted upstream over an ATM WAN VC. Different flavors of voice, full-range video, and video conferencing can be accommodated. The CE150 allows you to configure QoS using either the CopperView EM or the CopperCraft CLI. You must use three different MIB groups or tables:

- cmCircuitParam
- cmATMVcl
- cmParamSummary

You can create as many as 255 different subclasses of service using the cmCircuitParam group. You select either real time Variable Bit Rate (rtVBR) or non-real time Variable Bit Rate (nrtVBR), and then configure the subclasses with different sizes for Peak Cell Rate (PCR), Sustained Cell Rate (SCR), and Maximum Burst Size (MBS). You also can configure Unspecified Bit Rate (UBR).

Afterward, you can apply the different subclasses (or the different values for PCR, SCR, and MBS) to the different WAN VCs by using the cmATMVcl group.

Meanwhile, you can manage the WAN VCs and different subclasses of service by monitoring the total bandwidth assigned to ports using the cmParamSummary group. You can also adjust upward or downward subscription factors on the ports, allowing as much as 2500 percent oversubscription.

For more information about the MIB groups for applying QoS over ATM, see the MIB Definitions in the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual.

Configuring Quad T1 Frame Relay

The Quad T1 Frame Relay module, unlike the DS3 module for either Frame Relay or ATM, does not require the installation of software to run it. The software already resides on the System Control module in the CopperEdge.

The Quad T1 Frame Relay module is very similar to a DS3 Frame Relay module. The Quad T1 Frame Relay module has the same features and functions as a DS3 Frame Relay module, but the Quad T1 Frame Relay module has significantly lower throughput rates. Like the DS3 Frame Relay module, The Quad T1 Frame Relay module module will allow the configuration of VCs and the shaping of traffic using CIR on the VCs.

Performance Monitoring

Many interfaces and operation modes collect summary packet-counts useful in monitoring and evaluating link and interface performance. In most cases the counts are cumulative since the system was last booted. Counters can only be reset by resetting the CE150. For complete information on available performance monitoring groups and their objects, see the specific tables in the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual.

IP Monitoring with IfTable group

The IfTable is a MIB-II standard table where information on IP interfaces and their data performance is collected and stored. The IfTable includes *only* IP interfaces. Therefore, Frame Relay PVCs used only as policy-routed destinations do not appear in the table. PVCs that have IP addresses on either end appear in the IfTable, but *only IP-routed packets are counted; policy-routed packets are not counted in the IfTable*.

VC Monitoring with frCircuitTable

The frCircuit group is a standard Frame Relay MIB table with configuration and performance information about the configured PVCs. Information in this table is specific to Frame Relay and does not include data on IP addresses and routing.

Monitoring Frame Relay Errors with frErrTable

The frErrTable, also derived from RFC-1315, is a standard table of information about the most recent errored frame detected over a specified Frame Relay interface.

Disabling/Deprovisioning an Interface

To temporarily take any interface or port out of service, set the IfTable AdminStatus to Down. To restore service at the same level and to the same users, enter this command:

```
CRAFT> set iftable [pii] adminstatus=up
```

To completely *deprovision* a port (that is, remove it as an active interface or prepare it for a fresh configuration), set the cmIface table entry for the interface to NetModel=None, and clear the Name, Group, and AdditionalInfo objects.

Chapter 6
Voice over DSL

This chapter describes the procedures used to plan and configure networks for voice over SDSL and T1 line modules. It discusses the configuration of DSL and WAN interfaces using different netmodels along with, in the case of Integrated Access Devices (IADs), configuration of both voice and data subnets on a single port.

An IAD serving both voice and data circuits, in contrast to a CPE serving only data circuits, requires a dual netmodel and dual pathways over the IAD and the CE150.

Overview of VoDSL

In delivering digitized voice over SDSL or T1, the CE150 is able to work with a series of different IADs on local loops, and a series of different voice gateways on DS-1s, DS-3s, and OC-3s. At present, the CE150 is able to interface with voice gateways from a number of other manufacturers.

Signaling Protocols

Two signaling protocols for call setup and for voice communications over the CE150 exist: GR-303 and Media Gateway Control Protocol (MGCP). Some IADs are able to work only with the GR-303 protocol; others are able to work only the MGCP standard.

A few IADs, such as Copper Mountain's CR408, CR508, and CR508T, are able to work with gateways using either the GR-303 or the MGCP standard. But, if the CopperMountain IADs have code compatible with GR-303, you must download new code to make them compatible with MGCP, and vice versa.

Service Architecture

Two types of service architecture for carrying call setup messages and voice packets exist: ATM and IP. Some voice gateways as well as the IADs capable of working with them use an ATM AAL2-based architecture. In contrast, other voice gateways as well as the IADs capable of working with them use an IP-based architecture.

Dual Pathways on a Port

All IADs provide data and voice access to upstream networks for users downstream. But, although the data streams and voice streams go over the same DSL line to and from the CE150, the data and voice packets are separated into two VCs on DSL lines and on DS1 and DS3 lines. Two pathways on the WAN interface must be created. Two paths over the CE150 are necessary because the voice packets must go to their own voice gateway and the data packets must go to their own data gateway. The voice gateway performs call setup and routes voice packets; the data gateway reads header information and, in addition, routes data packets.

Dual Netmodels on a Port

To accommodate the dual pathways over DSL ports and WAN ports on the CE150, dual netmodels are necessary. For example, on data VCs on DSL ports, you can use any of five netmodels:

- IP
- VWAN
- CopperVPN
- Cross-Connect
- HDIA

On voice VCs on DSL ports, which must be set on VC 22, you can use either of two netmodels:

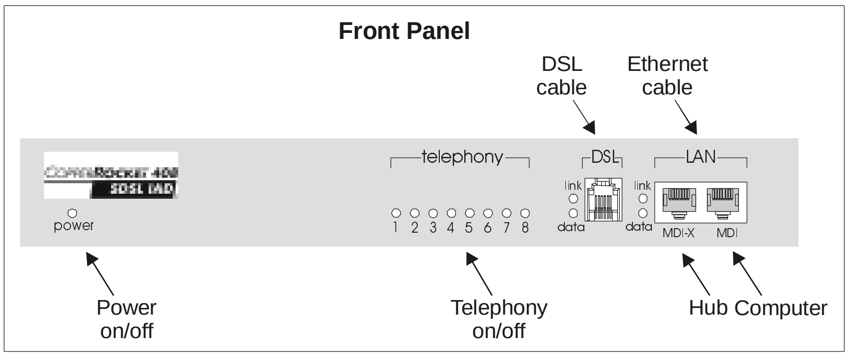
- Cross-Connect
- HDIA

The procedures for configuring data VCs on the DSL ports for the IP, VWAN, CopperVPN, and Cross-Connect netmodels are the same as they have always been. Since you must set the netmodel on the port, you do not need a VC. The system uses a default of 528. In contrast, the procedures for configuring voice VCs on the DSL ports for the HDIA netmodel are new and require some explanation. Special configuration of the IADs is necessary.

Dual Pathways on an IAD

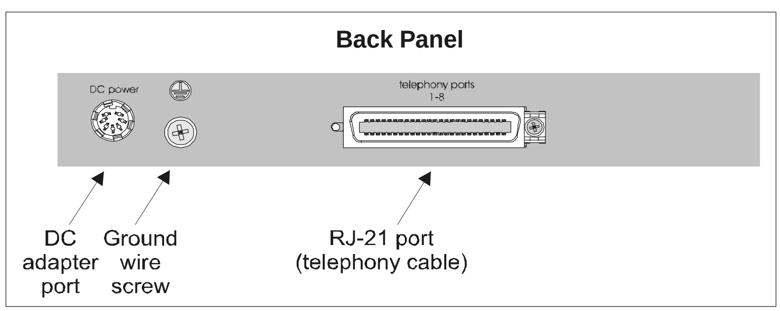
The CopperRocket IADs (CR408, CR508, CR508T), like all IADs, provide two types of access. They permit voice and data VCs on a single SDSL line, running at speeds ranging from 128 Kbps to 1.5 Mbps. For sending and receiving voice packets, the CopperRocket IADs are compatible with both GR-303 gateways and other types of gateways.

The front panel of a CR408, for example, has connectors for a DSL twisted pair and an Ethernet cable. With the 10/100BaseT connector, users on a LAN can access the data subnet on the CE150.



Front Panel of the CR408

The rear panel of a CR408 has an RJ-21 connector. With the RJ-21 connector, users on eight phone lines at a time can access the voice subnet on the CE150. In addition, for emergency service, a CR408 has a LifeLine feature that allows two regular phone lines to be attached to its RJ-21 connector.

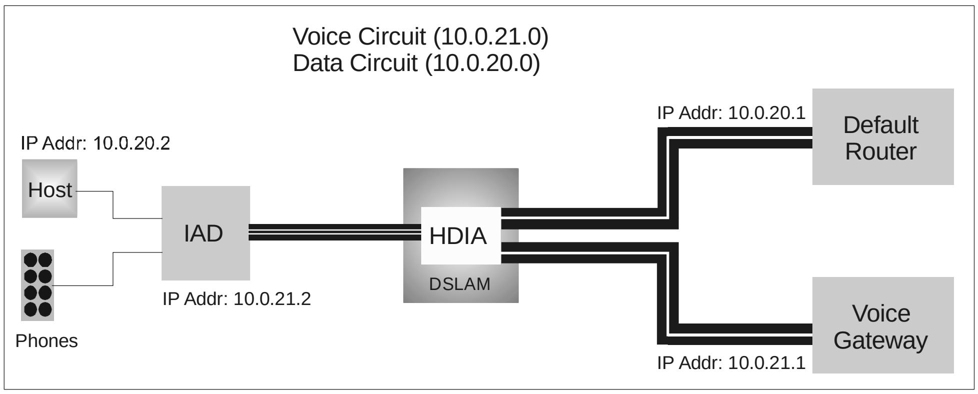


Back Panel of the CR408

HDIA Netmodel

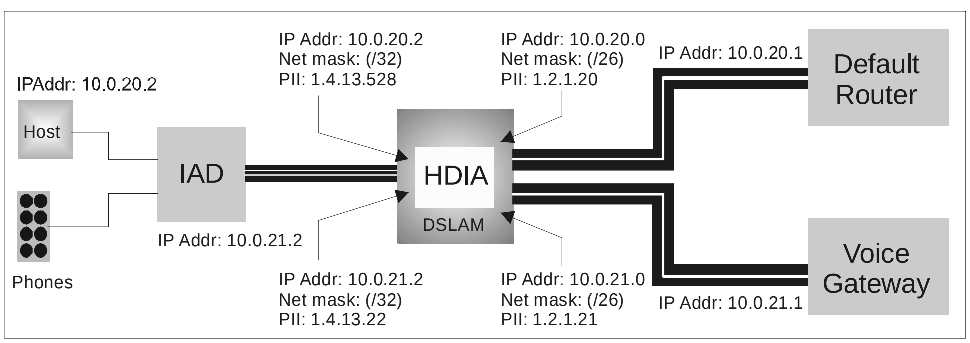
To manage separate voice and data circuits on DSL ports and WAN ports and to allow economical use of IP addressing, the CE150 provides a dual netmodel, called High Density IP Access (HDIA). It requires that you set up base IP addresses and address ranges (or net masks) for voice and data subnets on WAN ports. It also requires that you set up base IP addresses and address ranges (or net masks) for IADs and hosts on DSL ports.

HDIA allows you to place a router and hosts on different premises but on the same subnet and then assign to hosts a series of contiguous addresses by means of power-of-2 addressing boundaries. You must set the base address and address ranges on the DSL ports, but you must set the individual IP addresses on the hosts locally. Note in the two diagrams below that you must set up two HDIA groups: one for data and one for voice.



HDIA Netmodel—Data and Voice Networks

Each IAD has an IP address for its voice VC. Similarly, the hosts that the IAD serves through its Ethernet port have IP addresses on the data subnet. The data VCs on the subnet must be mapped to a single data VC on a WAN port. Also the data VC on the WAN port must be mapped to an upstream data router which itself must have an IP address on the subnet.



HDIA Netmodel—IP Addresses and Netmasks

To simplify the process of addressing and configuring CopperRocket IADs, the CE150 is able to exchange information with them through both CMCP and DHCP messages. The Plug and Play feature eliminates the need for users or administrators to configure CopperRocket IADs locally. The CE150 and the CopperRocket IADs attached to its DSL ports can send the following types of data:

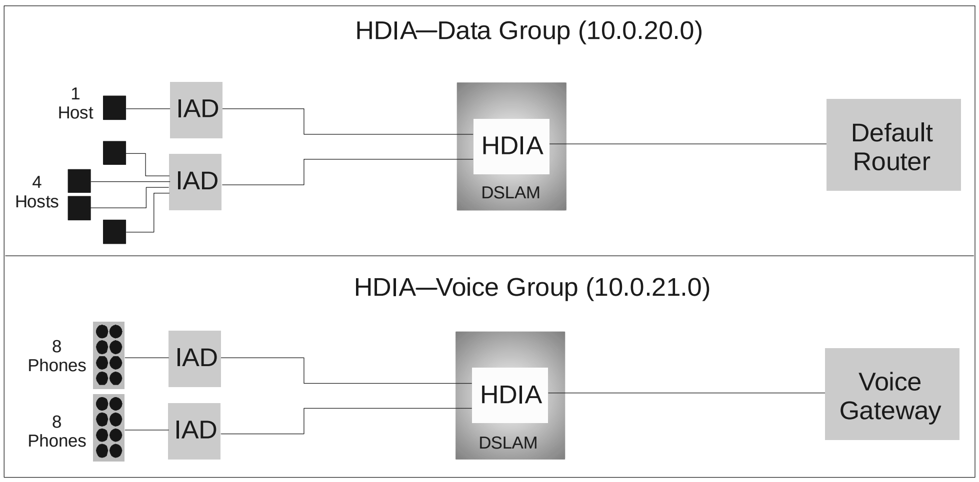
- IP addresses and netmasks for voice and data VCs
- voice gateway types; voice gateway IP addresses
- fragmentation capabilities
- maximum number of voice channels on CopperRocket IADs
- CPE data function and data encapsulation

Planning HDIA Data and Voice Subnets

High Density Internet Access (HDIA) uses an encapsulation type of IP-1490 on its DSL voice VCs and an encapsulation type of RFC-1483 on its DSL data VCs.

Multiple voice groups and multiple data groups are permissible. Different voice gateways located at different places can serve different voice groups; different ISPs or enterprise routers located at different places can serve different data groups.

In the following example, the focus is on IADs whose voice VCs belong to a single voice group and whose data VCs belong to a single data group. The data VCs on the DSL links have addresses on the same subnet as their router. The voice VCs have addresses on the same subnet as their voice gateway.



HDIA Netmodel—Data and Voice Groups

To plan the addressing for an HDIA data or voice group on the CE150, follow these steps:

1. DSLAM and IADs:
   a) Create two diagrams.

      Each has a DSLAM in the middle and multiple IADs on the left side connected by lines to the DSLAM.

   b) Label the first diagram, HDIA-Data.

   c) Label the second diagram, HDIA-Voice.
2. DSLAM, Router, and Voice Gateway:
   a) On the first diagram, draw a default router, connected to the right side of the DSLAM.

   b) On the second diagram, draw a voice gateway connected to the right side of the DSLAM.

3. Two IP Addressing Groups:

   The two diagrams show the data VCs and voice VCs through the same DSLAM.

   a) To distinguish the two groups, use one set of IP addresses, with a range of 10.0.20.1 to 10.0.20.63 (/26 subnet) for the data VCs.

   b) Use another set of IP addresses with a range of 10.0.21.1 to 10.0.21.63 (/26 subnet) for the voice VCs.

4. Default Router and Voice Gateway:

   a) Assign the 10.0.20.1 address to the default router on the data subnet.

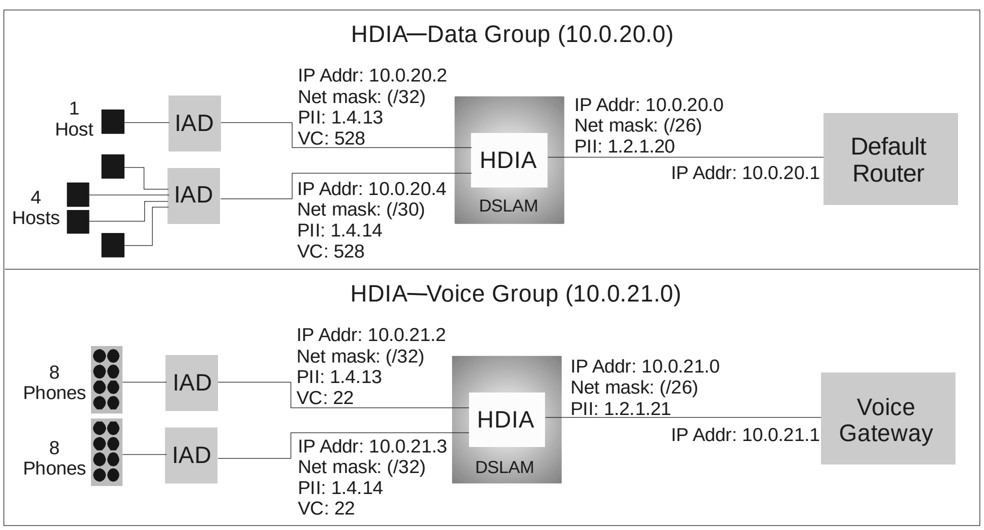
   b) Assign the 10.0.21.1 address to the voice gateway on the voice subnet.

5. IP Addresses and Subnet Masks on WAN Ports:

   a) Assign the 10.0.20.0 address with a subnet mask of 255.255.255.192 (or /26) to the data VC on the WAN port.

   b) Assign the 10.0.21.0 address with a subnet mask of 255.255.255.192 (or /26) to the voice VC on the WAN port.

   The two WAN VCs themselves, though, do not have IP addresses. In the data group, only the hosts and the default router have IP addresses. In the voice group, only the IADs and the voice gateway have IP addresses.



HDIA Netmodel—Data and Voice Groups with Netmasks

6. IP Addresses on the data DSL Ports:

   a) Assign the 10.0.20.2 address with a subnet mask of 255.255.255.255 (/32) to the first DSL port.

      Only one host can be attached to this IAD.

   b) Assign the 10.0.20.4 address with a subnet mask of 255.255.255.252 (/30) to the second DSL port.

      As many as four hosts can be attached to this IAD.

7. IP Addresses on the voice DSL Ports:

   a) Assign the 10.0.21.2 address with a subnet mask of 255.255.255.255 (/32) to the first DSL port.

      Only one host, or group of eight phones, can be attached to this IAD.

   b) Assign the 10.0.21.3 address with a subnet mask of 255.255.255.255 (/32) to the second DSL port.

      Only one host, or group of eight phones, can be attached to this IAD.

N O T E

The term, voice VC, refers to VC 22. More generally, VC 22 is the high-priority VC on the CE150*. It should be used for any delay-sensitive traffic, including voice and video. In the downstream direction, the* CE150 *gives priority to traffic on VC 22 over traffic on other VCs on the same DSL port.*

Planning Multiple Hosts on HDIA

Although the HDIA network model allows you to place multiple hosts on a data VC and assign them at different points throughout the subnet—theoretically, you have 256 options, since the numbers 0 and 255 are not reserved—you *cannot* choose any number at random. Restrictions exist.

First, all consecutive addresses on a single port or IAD must belong to a range that is a power of 2. The smallest range is 1 address. The next larger can have two addresses for two hosts, the next larger can have four, etc. A customer wanting three consecutive addresses for hosts must be assigned a range of four addresses. One will go unused. A customer wanting five addresses must be assigned a range of eight addresses. Three will go unused. Different ranges, of course, can exist on the same subnet.

Second, different ranges (2, 4, 8, 16, etc.) must be assigned to certain base numbers on the subnet. For example, a range of two addresses must be assigned to even base numbers on the subnet (that is, only numbers divisible by two). A range of four addresses must be assigned to base numbers on the subnet divisible by four; and a range of eight addresses must be assigned to base numbers on the subnet divisible by eight.

The following chart lists all of the base addresses on the subnet for each of the different ranges.

Comparison of Base Addresses and Ranges of Addresses

Hosts (IP Addresses) per Range	Number of Ranges in the Subnet	List of Base Addresses for the Ranges
/32 = 1 Host	256	all integers and 0
/31 = 2 Hosts	128	all even integers and 0
/30 = 4 Hosts	64	0, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60, 64, 68, 72, 76, 80, 84, 88, 92, 96, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140, 144, 148, 152, 156, 160, 164, 168, 172, 176, 180, 184, 188, 192, 196, 200, 204, 208, 212, 216, 220, 224, 228, 232, 236, 240, 244, 248, 252
/29 = 8 Hosts	32	0, 8, 16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128, 136, 144, 152, 160, 168, 176, 184, 192, 200, 208, 216, 224, 232, 240, 248
/28 = 16 Hosts	16	0, 16, 32, 48, 64, 80, 96, 112, 128, 144, 160, 176, 192, 208, 224, 240
/27 = 32 Hosts	8	0, 32, 64, 96, 128, 160, 192, 224
/26 = 64 Hosts	4	0, 64, 128, 192

The numbers in the chart also can be generated by applying the following rule: an IP address cannot have any ones where the subnet mask has zeros. For example, the address 10.0.21.6 cannot be assigned a subnet mask of 255.255.255.252 or /30 because the number 6 in binary format is 0000 0110 and the number 252 in binary format is 1111 1100.

The following table compares the numbers 1-32 in binary format with six different net masks in binary format. It shows what ranges can be assigned to base numbers.

Comparison of IP Addresses and Net Masks in Binary Format

IP Base Address	1 Host (/32) 1111	2 Hosts (/31) 1110	4 Hosts (/30) 1100 (/30)	8 Hosts (/29) 1000	16 Hosts (/28) 0000 (/28)	32 Hosts (/27) 0 0000
(.1) 0001	y	n	n	n	n	n
(.2) 0010	y	y	n	n	n	n
(.3) 0011	y	n	n	n	n	n
(.4) 0100	y	y	y	n	n	n
(.5) 0101	y	n	n	n	n	n
(.6) 0110	y	y	n	n	n	n
(.7) 0111	y	n	n	n	n	n
(.8) 1000	y	y	y	y	n	n
(.9) 1001	y	n	n	n	n	n
(.10) 1010	y	y	n	n	n	n
(.11) 1011	y	n	n	n	n	n
(.12) 1100	y	y	y	n	n	n
(.13) 1101	y	n	n	n	n	n
(.14) 1110	y	y	n	n	n	n
(.15) 1111	y	n	n	n	n	n
(.16) 0001 0000	y	y	y	y	y	n
(.17) 0001 0001	y	n	n	n	n	n
(.18) 0001 0010	y	y	n	n	n	n
(.19) 0001 0011	y	n	n	n	n	n
(.20) 0001 0100	y	y	y	n	n	n
(.21) 0001 0101	y	n	n	n	n	n
(.22) 0001 0110	y	y	n	n	n	n
(.23) 0001 0111	y	n	n	n	n	n
(.24) 0001 1000	y	y	y	y	n	n
(.25) 0001 1001	y	n	n	n	n	n
(.26) 0001 1010	y	y	n	n	n	n
(.27) 0001 1011	y	n	n	n	n	n
(.28) 0001 1100	y	y	y	n	n	n
(.29) 0001 1101	y	n	n	n	n	n
(.30) 0001 1110	y	y	n	n	n	n
(.31) 0001 1111	y	n	n	n	n	n
(.32) 0010 0000	y	y	y	y	y	y

Configuring HDIA Data and Voice Subnets

The procedure for setting up addressing on HDIA data and voice subnets on the CE150 is different from the procedure for planning them. In planning, it is best to start with the hosts and IADs at customer sites. In setting up the addressing, it is best to start with the WAN VCs and the data and voice gateways. Then, work down to the DSL ports, assigning the proper number of VCs on all of the ports to accommodate all of the hosts and IADs.

In the following example, the focus is on a single voice subnet and data subnet and a single IAD on the two subnets. But the procedure would be the same for installing multiple IADs with multiple hosts attached to each IAD. Different sized groups of hosts, however, require different subnet masks. In this example, you will see only a single host on the IAD. So the subnet mask will be 255.255.255.255 (or /32).

Also, to assign IP addresses to hosts on the data subnet, you can do it manually or automatically (dynamically), using a DHCP server located upstream from the CE150. This procedure shows you how to assign the IP addresses with the DHCP server. It also shows you how you how to assign the IP address for the voice VC automatically (dynamically) by using the CE150 as a DHCP server.

To set up addressing for an HDIA data and voice group on the CE150, follow these steps:

1. Set the VC for data on a WAN port, using the address 10.0.20.0, and using a 255.255.255.192 subnet mask (/26):

   ```
   CRAFT> set cmiface [1.2.1.20] netmodel=hdia
   ipaddr=10.0.20.0 netmask=255.255.255.192
   encapsulationtype=rfc1483 farendaddr=10.0.20.1
   ```

   The FarEndAddr is the data router for the data group. It should be on the same subnet as host IP addresses. When you perform a Get on the VC, you will see this table:

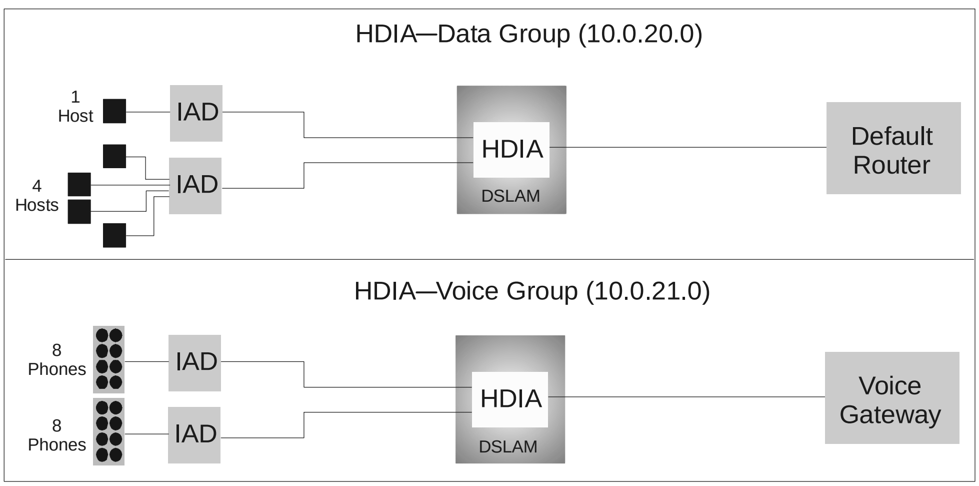
   ```
   CRAFT> get cmiface [1.2.1.20]
   Group: cmIfaceTable
   Instance: [1.2.1.20]
   PII                  =  1.2.1.20
   IfIndex              =  1.2.1.20
   Name                 =  ""
   GroupName            =  ""
   AdditionalInfo       =  ""
   NetModel             =  HDIA
   IpAddr               =  10.0.20.0
   NetMask              =  255.255.255.192
   MacAddr              =  ff.ff.ff.ff.ff.ff
   BurnedInMacAddr      =  ff.ff.ff.ff.ff.ff
   FarEndAddr           =  10.0.20.1
   DestPII              =  0.0.0.0
   CMCPCompatible       =  No
   EncapsulationType    =  rfc1483
   FwdMode              =  HDIA
   Pix                  =  532
   ServiceClass         =  None
   ```

2. Set the VC for voice on a WAN VC, using the address 10.0.21.0, and using a 255.255.255.192 subnet mask (/26):

```
CRAFT> set cmiface [1.2.1.21] netmodel=hdia
ipaddr=10.0.21.0 netmask=255.255.255.192
encapsulationtype=rfc1483 farendaddr=10.0.21.1
```

The FarEndAddr is the voice gateway for the voice group. It must be on the same subnet as the IAD IP addresses. The following diagram shows addressing for both voice and data subnets on WAN VCs on a single WAN port. When you perform a Get on the VC to check your work, you will see this table:

```
CRAFT> get cmiface [1.2.1.21]
Group: cmIfaceTable
Instance: [1.2.1.21]
PII                    =  1.2.1.21
IfIndex                =  1.2.1.21
Name                   =  ""
GroupName              =  ""
AdditionalInfo         =  ""
NetModel               =  HDIA
IpAddr                 =  10.0.21.0
NetMask                =  255.255.255.192
MacAddr                =  ff.ff.ff.ff.ff.ff
BurnedInMacAddr        =  ff.ff.ff.ff.ff.ff
FarEndAddr             =  10.0.21.1
DestPII                =  0.0.0.0
CMCPCompatible         =  No
EncapsulationType      =  rfc1483
FwdMode                =  HDIA
Pix                    =  533
ServiceClass           =  None
```



HDIA Netmodel—Configuring Subnets for Data and Voice Groups

3. Set the VC for data on the WAN port with a vpi (range=0 to 15) and a vci (range=0 to 511). But, reserve the vpi=0 and vci=511 combination for ATM loopback testing.

```
CRAFT> set cmatmvcl [1.2.1.20] rowstatus=create vpi=0
vci=200 adminstatus=up
```

When you perform a Get on the VC to check your work, you will see this table:

```
CRAFT> get cmatmvcl [1.2.1.20]
Group: cmAtmVclTable
Instance: [1.2.1.20]
PII                   =  1.2.1.20
Vpi                   =  0
Vci                   =  200
AdminStatus           =  Up
OperStatus            =  Up
LastChange            =  0 day 0 hour 0 min 0.0 sec
AalType               =  Aal5
Aal5CpcsTransmitSduS =  1600
Aal5CpcsReceiveSduSi =  1600
RowStatus             =  Active
TransmitTrafficDescr =  1
OAMState              =  adminDown(0)
OAMAdminState         =  disabled
OAMAutoLBState        =  disabled
OAMManualLBCmd        =  none
OAMManualLBCmdStatus =  none
OAMLBInterval         =  5
OAMLBTimeOut          =  15
OAMTxAISCells         =  0
OAMRxRDICells         =  0
OAMRxAISCells         =  0
OAMTxRDICells         =  0
OAMTxLBRequestCells  =  0
OAMRxLBResponseCells =  0
OAMRxLBRequestCells  =  0
OAMTxLBResponseCells =  0
OAMRxUnsupportedCell =  0
OAMTxDiscards         =  0
OAMRxDiscards         =  0
```

4. (Optional) If you do not want to set the data subnet for peer-to-peer traffic, disable the VC for data on the WAN port. (The default is enabled.)

```
CRAFT> set cmvpngroup [1.2.1.20] peertopeer=disabled
```

Unless you have a specific reason for disabling peer-to-peer traffic, we recommend you leave this option enabled.

5. Set the VC for voice on the WAN port with a VPI (range=0-15) and a VCI (range=0-511). But, reserve the vpi=0 and vci=511 combination for ATM loopback testing.

   ```
   CRAFT> set cmatmvcl [1.2.1.21] rowstatus=create vpi=0
   vci=201 adminstatus=up
   ```

   When you perform a Get on the VC to check your work, you will see this table:

   ```
   CRAFT> get cmatmvcl [1.2.1.21]
   Group: cmAtmVclTable
   Instance: [1.2.1.21]
   PII                  =  1.2.1.21
   Vpi                  =  0
   Vci                  =  201
   AdminStatus          =  Up
   OperStatus           =  Up
   LastChange           =  0 day 0 hour 0 min 0.0 sec
   AalType              =  Aal5
   Aal5CpcsTransmitSduS =  1600
   Aal5CpcsReceiveSduSi =  1600
   RowStatus            =  Active
   TransmitTrafficDescr =  1
   TransmitTrafficDescr =  2
   OAMState             =  adminDown(0)
   OAMAdminState        =  disabled
   OAMAutoLBState       =  disabled
   OAMManualLBCmd       =  none
   OAMManualLBCmdStatus =  none
   OAMLBInterval        =  5
   OAMLBTimeOut         =  15
   OAMTxAISCells        =  0
   OAMRxRDICells        =  0
   OAMRxAISCells        =  0
   OAMTxRDICells        =  0
   OAMTxLBRequestCells  =  0
   OAMRxLBResponseCells =  0
   OAMRxLBRequestCells  =  0
   OAMTxLBResponseCells =  0
   OAMRxUnsupportedCell =  0
   OAMTxDiscards        =  0
   OAMRxDiscards        =  0
   ```

6. Set the VC for voice on the WAN port with a voice gateway type ID and a voice gateway IP address. The voice gateway IP address must be on the same subnet as the IAD IP addresses:

   ```
   CRAFT> set cminterfaceoptions [1.2.1.21]
   voicegatewaytypeid=4 voicegatewayipaddr=10.0.21.1
   ```

   When you perform a Get on the VC to check your work, you will see this table. Do *not* try to set JitterSpeedTradeOff:

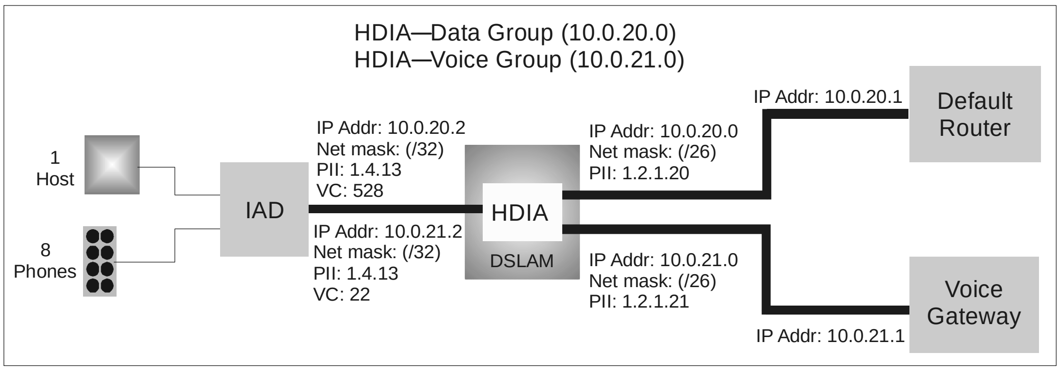
   ```
   CRAFT> get cminterfaceoptions [1.2.1.21]
   Group: cmInterfaceOptionsTable
   Instance: [1.2.1.21]
   PII                  =  1.2.1.21
   VoiceGatewayType     =  TollBridge
   VoiceGatewayIpAddr   =  10.0.21.1
   MaxVoiceChannels     =  255
   JitterSpeedTradeoff  =  None
   JitterControlStatus  =  Uncontrolled
   RowStatus            =  Active
   VoiceGatewayTypeId   =  4
   ```

7. Set the VC for data on the first DSL port (1.4.13). Do not specify a number for the VC. The software uses the default of VC 528.

```
CRAFT> set cmiface [1.4.13] netmodel=hdia
ipaddr=10.0.20.2 netmask=255.255.255.255
destpii=1.2.1.20 encapsulationtype=rfc1483
```

The current netmask means only one host can be attached to the IAD. To attach four hosts, use a 255.255.255.252 netmask.

The following diagram shows addressing for data and voice VCs on the first DSL port.



HDIA Netmodel—Configuring IP Addresses and Netmasks

When you perform a get, you will see this table. Make sure the object, CMCPCompatible, shows a setting of Yes. If it does not, perform another Set command with CMCPCompatible=Yes.

```
CRAFT> get cmiface [1.4.13]
Group: cmIfaceTable
Instance: [1.4.13.0]
PII                    =  1.4.13.0
IfIndex                =  1.4.13.0
Name                   =  ""
GroupName              =  ""
AdditionalInfo         =  ""
NetModel               =  HDIA
IpAddr                 =  10.0.20.2
NetMask                =  255.255.255.255
MacAddr                =  0.60.58.1.43.12
BurnedInMacAddr        =  0.60.58.1.43.12
FarEndAddr             =  0.0.0.0
DestPII                =  1.2.1.20
CMCPCompatible         =  Yes
EncapsulationType      =  rfc1483
FwdMode                =  HDIA
Pix                    =  15
ServiceClass           =  D
```

8. Set the VC for data for DHCP. You are configuring the CE150 so that it can dynamically assign an IP address to the host on the data VC. You are assuming the existence of a DHCP server upstream from the CE150 and its default router.

```
CRAFT> set cmdhcp [1.4.13.22] ipaddress=10.0.21.2
netmask=255.255.255.255 function=dhcpforward
serveripaddr=10.0.25.2
```

   When you perform a get, you will see this table:

```
CRAFT> get cmdhcp [1.4.13.22]
Group: cmDHCPTable
Instance: [1.4.13.22]
PII                 =  1.4.13.22
RowStatus           =  Active
IpAddress           =  10.0.20.2
NetMask             =  255.255.255.0
DefaultRouter       =  0.0.0.0
DNSServer           =  0.0.0.0
Function            =  DHCPForward
ServerIPAddr        =  10.0.25.2
CircuitID           =  CRAFT-1.4.13.22
```

9. Set the VC for voice on the DSL port (1.4.13.22). You must specify the number 22 for a VC. You *cannot* use any VC for voice except VC 22. You cannot use any netmask for voice other than 255.255.255.255.

```
CRAFT> set cmiface [1.4.13.22] netmodel=hdia
ipaddr=10.0.21.2 netmask=255.255.255.255
destpii=1.2.1.21 encapsulationtype=ip-1490
```

   When you perform a get, you will see this table. Once again, make sure the object, CMCPCompatible, shows a setting of Yes. If it does not, perform another Set command with CMCPCompatible=Yes.

```
CRAFT> get cmiface [1.4.13.22]
Group: cmIfaceTable
Instance: [1.4.13.22]
PII                 =  1.4.13.22
IfIndex             =  1.4.13.22
Name                =  ""
GroupName           =  ""
AdditionalInfo      =  ""
NetModel            =  HDIA
IpAddr              =  10.0.21.2
NetMask             =  255.255.255.255
MacAddr             =  ff.ff.ff.ff.ff.ff
BurnedInMacAddr     =  ff.ff.ff.ff.ff.ff
FarEndAddr          =  0.0.0.0
DestPII             =  1.2.1.21
CMCPCompatible      =  Yes
EncapsulationType   =  IP-1490
FwdMode             =  HDIA
Pix                 =  2590
ServiceClass        =  None
```

10. Set the DSL port (1.4.13) for the maximum number of voice channels. In this case, the IAD is capable of a maximum of eight channels. You can set JitterSpeedTradeoff on the CE150 for values other than FullSpeedDataOnly, which is the default. Currently, the most appropriate setting for ports with a voice VC on them is MinJitterAndSpeed.

```
CRAFT> set cminterfaceoptions [1.4.13]
maxvoicechannels=8 jitterspeedtradeoff=
minjitterandspeed
```

When you perform a get, you will see this table:

```
CRAFT> get cminterfaceoptions [1.4.13]
Group: cmInterfaceOptionsTable
Instance: [1.4.13.0]
PII                  =  1.4.13.0
VoiceGatewayType     =  None
VoiceGatewayIpAddr   =  0.0.0.0
MaxVoiceChannels     =  8
JitterSpeedTradeoff  =  MinJitterAndSpeed
JitterControlStatus  =  Uncontrolled
RowStatus            =  Active
VoiceGatewayTypeId   =  1
```

11. Set the VC for voice for DHCP. You are configuring the CE150 so that it, not an upstream DHCP server, can dynamically assign an IP address to the IAD on the voice VC.

```
CRAFT> set cmdhcp [1.4.13.22] ipaddress=10.0.21.2
netmask=255.255.255.255 function=dhcprespond
```

When you perform a get, you will see this table:

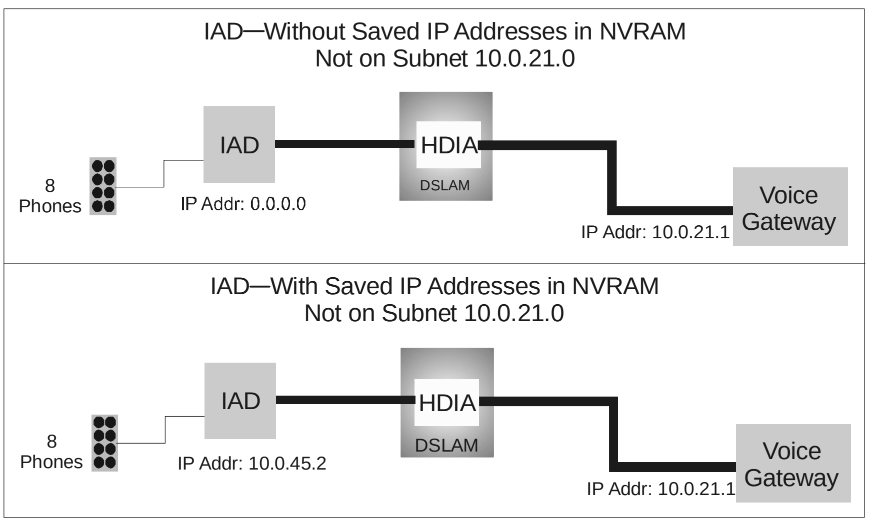
```
CRAFT> get cmdhcp [1.4.13.22]
Group: cmDHCPTable
Instance: [1.4.13.22]
PII                  =  1.4.13.22
RowStatus            =  Active
IpAddress            =  10.0.21.2
NetMask              =  255.255.255.0
DefaultRouter        =  0.0.0.0
DNSServer            =  0.0.0.0
Function             =  DHCPRespond
ServerIPAddr         =  0.0.0.0
CircuitID            =  CRAFT-1.4.13.22
```

12. Perform a Get on the IAD (CPE:1.4.13) attached to the first DSL port. See if the CurrentIPAddr and SavedIpAddr and the CurrentCAIpAddr and SavedCAIpAddr are zeros or non-zeros. If they are zeros as shown in the following example, go to step 14.

```
CRAFT> get cmcpeiad [cpe:1.4.13]
Group: cmCpeIADTable
Instance: [CPE:1.4.13]
Index                =  CPE:1.4.13
CurrentIpAddr        =  0.0.0.0
CurrentCAIpAddr      =  0.0.0.0
SavedIpAddr          =  0.0.0.0
SavedCAIpAddr        =  0.0.0.0
NumVoicePorts        =  0
NumConnections       =  0
Command              =  None
TOSByte              =  0
LogAction            =  None
LogType              =  ALL(0)
LogLevel             =  INFO(0)
LogDuration          =  30
TotalMsgsLogged      =  0
Trace1Mask           =  0
Trace2Mask           =  0
```

If the IP addresses are already set and they are different from the ones set in steps 5 to 9, go to step 13.

```
CRAFT> get cmcpeiad [cpe:1.4.13]
Group: cmCpeIADTable
Instance: [CPE:1.4.13]
Index                =  CPE:1.4.13
CurrentIpAddr        =  10.0.45.2
CurrentCAIpAddr      =  10.0.45.1
SavedIpAddr          =  10.0.45.2
SavedCAIpAddr        =  10.0.45.1
NumVoicePorts        =  8
NumConnections       =  8
Command              =  None
TOSByte              =  0
LogAction            =  None
LogType              =  ALL(0)
LogLevel             =  INFO(0)
LogDuration          =  30
TotalMsgsLogged      =  0
Trace1Mask           =  0
Trace2Mask           =  0
```



HDIA Netmodel—Configuring the IP Address on the CR408

13. If the IP addresses are already set and are different from the ones set in steps 5-9, set either of the two saved addresses on the IAD (CPE:1.4.13) to zero.

```
CRAFT> set cmcpeiad [cpe:1.4.13]
command=savedipaddrerase
```

When you perform a Get**,** you will see the following:

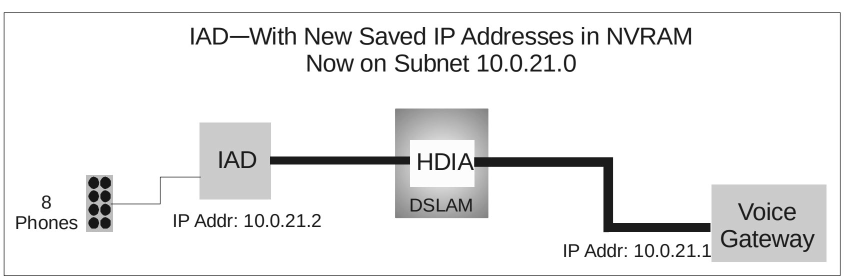
```
CRAFT> get cmcpeiad [cpe:1.4.13]
Group: cmCpeIADTable
Instance: [CPE:1.4.13]
Index                =  CPE:1.4.13
CurrentIpAddr        =  10.0.45.2
CurrentCAIpAddr      =  10.0.45.1
SavedIpAddr          =  0.0.0.0
SavedCAIpAddr        =  0.0.0.0
NumVoicePorts        =  8
NumConnections       =  8
Command              =  None
TOSByte              =  0
LogAction            =  None
LogType              =  ALL(0)
LogLevel             =  INFO(0)
LogDuration          =  30
TotalMsgsLogged      =  0
Trace1Mask           =  0
Trace2Mask           =  0
```

14. Restart the IAD. When the IAD reboots, it asks the CE150 for a new IP address and new IP address for its voice gateway. To restart or retrain, use cmCpeBoard (command=restart). To retrain, use ifTable (adminstatus =down, adminstatus=up). Failure to restart/retrain means the IAD to use its old addresses.

```
CRAFT> set iftable [1.4.13] adminstatus=down
CRAFT> set iftable [1.4.13] adminstatus=up
```

When you perform a Get, you will see this table.

```
CRAFT> get cmcpeiad [cpe:1.4.13]
Group: cmCpeIADTable
Instance: [CPE:1.4.13]
Index                =  CPE:1.4.13
CurrentIpAddr        =  10.0.21.2
CurrentCAIpAddr      =  10.0.21.1
SavedIpAddr          =  10.0.21.2
SavedCAIpAddr        =  10.0.21.1
NumVoicePorts        =  8
NumConnections       =  8
Command              =  None
TOSByte              =  0
LogAction            =  None
LogType              =  ALL(0)
LogLevel             =  INFO(0)
LogDuration          =  30
TotalMsgsLogged      =  0
Trace1Mask           =  0
Trace2Mask           =  0
```

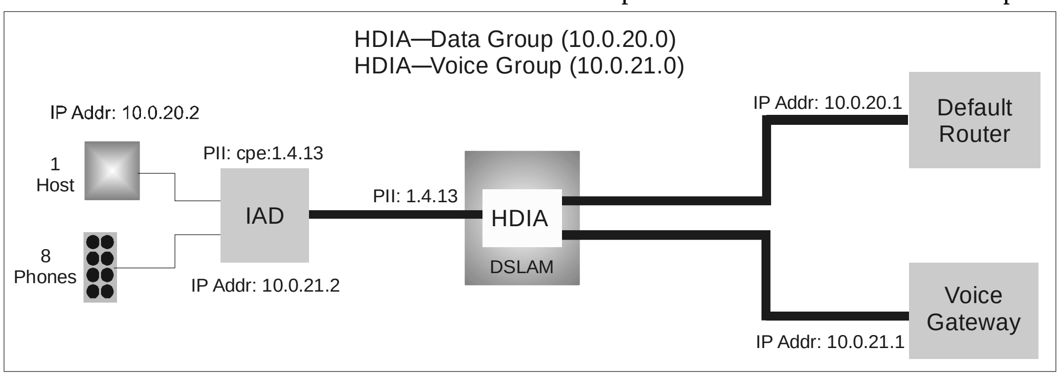


HDIA Netmodel—Restarting the IAD

15. Perform a Get on the board in the IAD (cpe: 1.4.13) and confirm that it is an IAD capable of supporting both data and voice. The software version should be the latest version. The file name should always be CPE5_T.BIN.

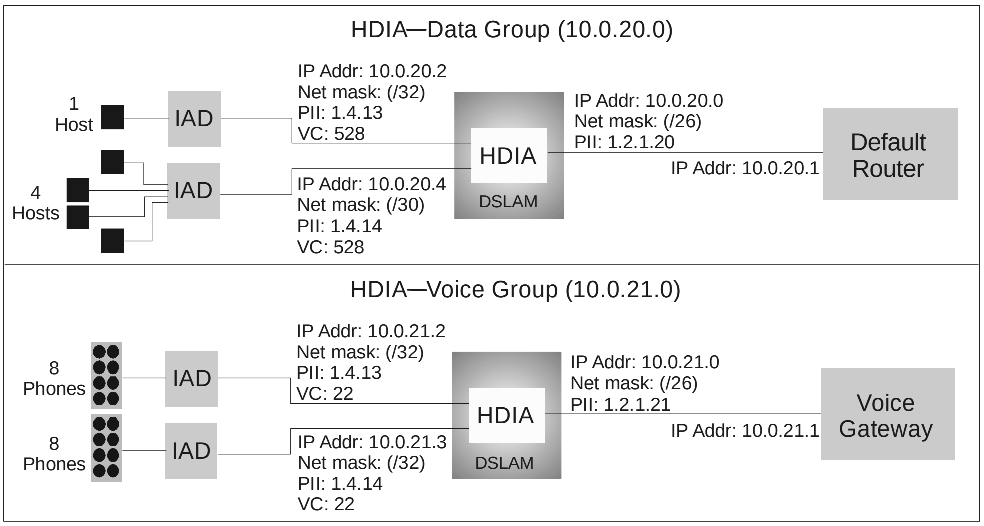
```
CRAFT> get cmcpeboard [cpe:1.4.13]
Group: cmCPEBoardTable
Instance: [CPE:1.4.13]
Index                =  CPE:1.4.13
ObjectClass          =  CPE-SDSL
OperState            =  Enabled
HwType               =  CPE-SDSL
HwVersion            =  R 1.1
SwVersion            =  3.0.79
PromVersion          =  3.0.70
Role                 =  Active
UpTime               =  0 day 3 hour 6 min 1.0 sec
                        (2000/02/15-21:37:40)
NumPorts             =  1
FileName             =  CPE5_T.BIN
FileDate             =  Feb 14 2000, 11:46:31
ConfigChange         =  0
Command              =  None
SerialNumber         =  0.60.58.1.43.12
VendorDescription    =  CR408 - SDSL IAD (8 Port)
GroupMap             =  3a.40.39.38.91.92.93.94.9b
ManagementOptions    =  1
```

The following diagram shows the addressing on both the data host and the voice IAD as well as the default router for data and the voice gateway. It also shows the PII of the first DSL port and the IAD on the DSL port.



HDIA Netmodel—Querying the CR408

16. Set the addressing for the second DSL port and the second IAD. The subnets for the data and voice groups have been set on the WAN port. Data and voice VCsalso have been configured on the WAN port. The second IAD is going to support four hosts, as shown in the following diagram. Its subnet mask must be set at 255.255.255.252 (/30). The first of the four addresses can be 10.0.20.4; it cannot be 10.0.20.3.



HDIA Netmodel—Addressing the Second DSL Port

Chapter 7
ADSL: G.lite and G.dmt

This chapter describes the procedures used to plan and configure networks for data over ADSL modules, both G.lite and full-rate G.dmt. It discusses the configuration of DSL and WAN interfaces using different netmodels along with the setup and configuration of CPEs in router or in bridge mode.

The procedures include setting up virtual circuits on both the WAN and DSL interfaces of the CE150 with VPIs and VCIs. They do not include setting up virtual circuits on Telco routers on the WAN side of the network, or on CPE routers on the DSL side of the network.

Overview of G.dmt and G.lite

Both the G.dmt and G.lite modules have 24 ports. For G.lite on the downstream, each of the 24 ports supports multiples of 32 Kbps all the way from 64 Kbps to 2.336 Mbps. On the upstream, each of the ports supports multiples of 32 Kbps from 32 Kbps to 512 Kbps.

For G.dmt, on the downstream, each of the 24 ports supports multiples of 32 Kbps all the way from 32 Kbps to 6.144 Mbps. On the upstream, each of the ports supports multiples of 32 Kbps from 32 Kbps to 640 Kbps.

Configuration of the ports can be done either by the CopperView EM or by the CopperCraft command line interface. Using the cmAdslModem group, you can set parameters for upstream and downstream signal-to-noise margins and upstream and downstream receive and transmit rates.

CPEs Supporting G.dmt and G.lite

Both the G.dmt and G.lite modules are compatible with a variety of CPEs, including models from Alcatel, Lucent, Motorola, 3Com, Intel, Cisco, FlowPoint, Cayman, Efficient, and Westell. The G.dmt and G.lite modules are also capable of communicating with CPEs in either router or bridge mode. However, some CPEs will run only in bridge mode.

Netmodels Supporting G.dmt and G.lite

Both the G.dmt and G.lite modules are compatible with all of the current netmodels in the CE150, including IP, VWAN, CopperVPN, Cross-Connect, and HDIA. However, the LAN extension netmodels (IP, VWAN, CopperVPN, and HDIA-data) require the CPEs on the DSL links to be in bridge mode, not router mode. None of the CPEs is CMCP compatible; they must have the CMCPCompatible object in the cmIface group set to No.

Encapsulations Supporting G.dmt and G.lite

Both the G.dmt and G.lite modules support RFC-1483 for DSL port encapsulation. They also support RFC-2364 for DSL port encapsulation, but the CE150 allows only RFC-2364 encapsulation with VC-VC forwarding on the DS3 ATM WAN port.

When using RFC-2364 for DSL line encapsulation, the CE150 will not allow RFC-1973 encapsulation with PPP Translation forwarding on the DS3 ATM WAN port. It also will not allow RFC-2364 encapsulation with PPP-Transparent forwarding or with PPP-Translation forwarding on the DS3 ATM WAN port.

In addition, when using FRF.8, the CE150 will not allow a translation from RFC-1483 for DSL port encapsulation to an RFC-1490 encapsulation on the DS3 Frame Relay port.

VCs Supporting G.dmt and G.lite

Both the G.dmt and G.lite modules support multiple VCs on their DSL ports. The CE150 will allow you to map the VCs on the DSL ports to multiple VCs on its WAN VCs. The WAN VCs can be on DS3 ATM boards, DS3 Frame Relay boards, or DS1 boards. The number of VCs available are:

- On G.dmt and G.lite ports, the CE150 will support as many as eight VCs. In addition, the VPIs and VCIs on each VC on the ADSL ports must match the VPIs and VCIs on the CPE.
- On the WAN ports, the CE150 will support as many as 976 VCs. In addition, the VPIs and VCIs on each VC on an ATM board must match the VPIs and VCIs on the upstream router.

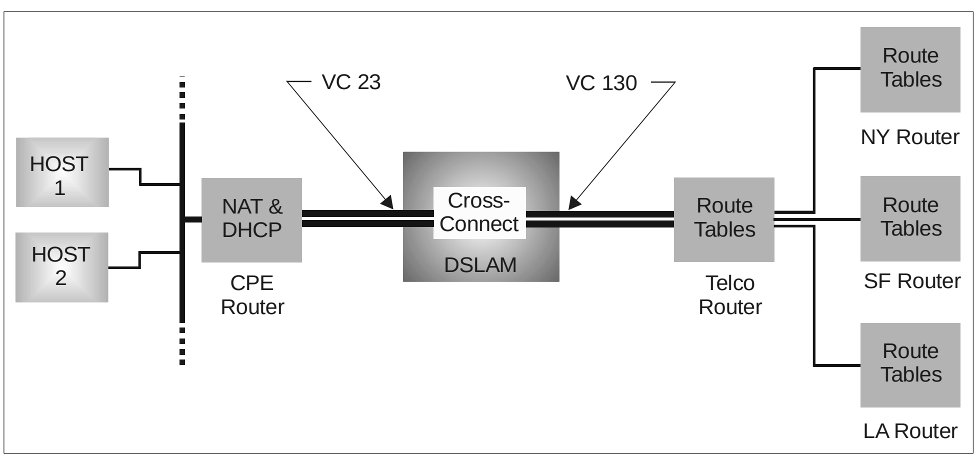
Planning the G.dmt and G.lite Network

Although the ADSL modules (both G.dmt and G.lite) support all netmodels, we will discuss only two of them: Cross-Connect in Point-to-Point Protocol (PPP) and Full IP. The two netmodels have different capabilities and different requirements for configuring CE150s and CPEs.

Cross-Connect Netmodel and Point-to-Point Protocol over ATM

If you have a LAN and a series of hosts downstream from your CPE and if you have it configured in bridge mode, the Cross-Connect netmodel can be set up with either public or private IP addresses for the hosts. In bridge mode, the CPE treats the hosts on the LAN as extensions of a subnet configured at the other end of the PPP connection. But, if you have a LAN and hosts downstream from your CPE and if you have it configured in router mode, the setup is different. A CPE in router mode can function as a simple or a complex router, using a DHCP function and a Network Address Translation (NAT) function.

In simple router mode, the setup typically requires a public IP address for the CPE and the hosts. In complex router mode, the setup typically requires an IP public address for the CPE. Using a DHCP function, the CPE then dynamically assigns private IP addresses to hosts on its LAN. When the hosts send messages upstream, the CPE sets up a table, recording all source and destination addresses. Then, before the CPE forwards packets upstream to a destination, it substitutes its own public address for the hosts' private addresses. In response, when messages come downstream for hosts on the LAN, the CPE consults its routing table for source and destination addresses. Before it forwards packets downstream, it substitutes the hosts' private addresses for its own public address.



Cross-Connect Netmodel and Point-to-Point Protocol

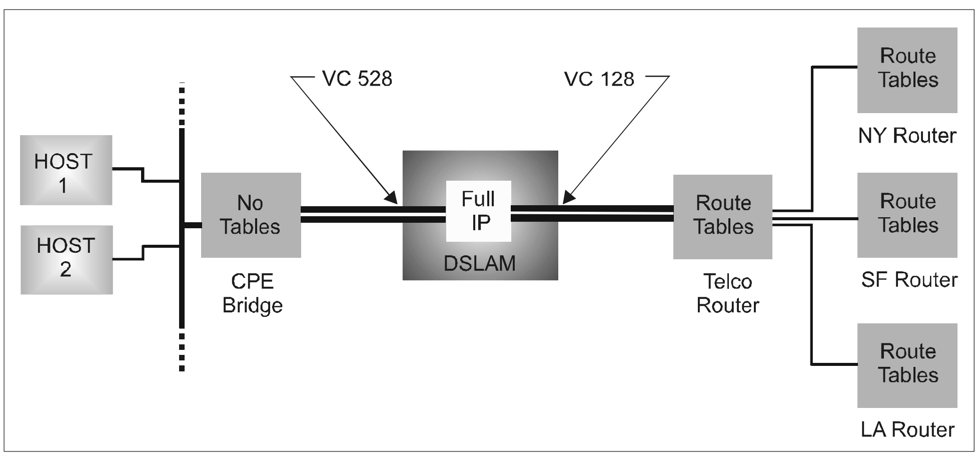
IP Netmodel

If you have a LAN and a series of hosts downstream from your CPE and if you have it configured in router mode, the IP netmodel can be set up with public IP addresses or private IP addresses for the hosts. In router mode, the CPE treats all of the hosts on the LAN as extensions of a subnet configured on the CE150.

But, if you have a LAN and a series of hosts downstream from your CPE and if you have it configured in bridge mode, the setup is different. A CPE in bridge mode also can be set up with either public or private IP addresses for hosts.

In bridge mode, hosts on a LAN use a DSL port on the CE150 as a gateway. Once packets arrive at a DSL port, the CE150 looks up its default IP route and sends them to a WAN VC. It forwards them over a Point-to-Point link to a Telco router. If the packets have been sent by hosts with public addresses, the router sends them out over the Internet to a destination address with their source address embedded.

If the packets have been sent by hosts with private addresses, the Telco router uses its NAT function and replaces the private addresses with its own public address. Then the router sends the packets out over the Internet to their destination with its public address (or source address) embedded. In response, when messages come downstream for hosts on the LAN, the router looks through its routing tables for the private addresses (or destination addresses). Before the router forwards the packets downstream, it replaces its own public address with the appropriate private addresses for the hosts.



IP Netmodel

Configuring the G.dmt and G.lite Data Network

The procedure for configuring the CE150 and CPEs for a Cross-Connect netmodel with PPP is different from the procedure for configuring them for a full IP netmodel. But, in both cases, it is best to start with the WAN VCs and their upstream routers. Then, work your way down to the DSL ports and link them to the WAN VCs.

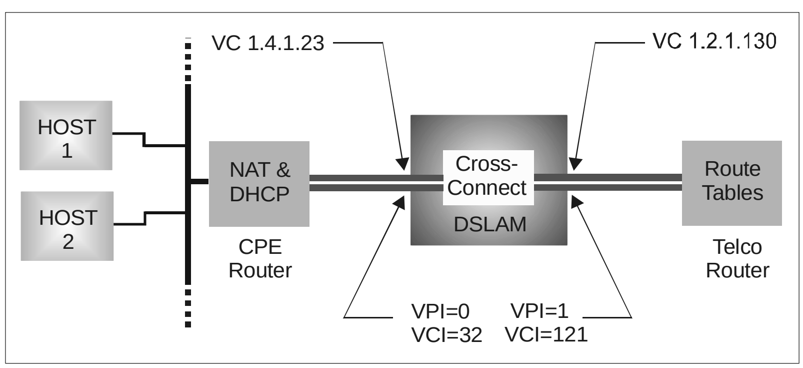
Finally, configure the CPE routers or CPE bridges and the hosts connected to them. The procedures for configuring the CPE routers or CPE bridges and the attached hosts, however, will vary according to the type and brand of CPE router or bridge and the type of operating system running on the host. But several basic steps are always present.

Cross-Connect Netmodel

In the Cross-Connect netmodel with PPP, the end points of the VCs must be manually configured for each stage in the link. (See the diagram below.)

VCs on the Telco router going to the CE150 must be mapped to VCs on the WAN interface of the CE150. WAN VCs on one side of the DSLAM must be mapped to the DSL VCs on the other side. And, finally, VCs on the DSL interface going to the CPE router must be mapped to VCs on the CPE. For each PPP link, the procedure must be repeated. A new WAN VC must be created for each new DSL VC.

In the following diagram, only one DSL port and one WAN port appear. In the procedure that follows, first the WAN VC, at 1.2.1.130, will be created. Then the DSL VC, at 1.4.1.23, will be created. Finally, the DSL VC will be mapped to the WAN VC. Configuration of the VC on the Telco router and configuration of the VC on the CPE router are not shown.



Cross-Connect Netmodel—VPIs and VCIs

For the first PPP connection on the DSL port, follow these steps:

1. Set the VC on the WAN port. The VPI and VCI on the WAN VC must match the ones on the Telco router, but the VCL does not have to match its counterpart on the Telco router. Set AdminStatus to Up.

   ```
   CRAFT> set cmatmvcl [1.2.1.130] vpi=1 vci=121
   adminstatus=up
   ```

   Perform a Get on the VC and you will see this table:

   ```
   CRAFT> get cmatmvcl [1.2.1.130]
   Group: cmAtmVclTable
   Instance: [1.2.1.130]
   PII                   =  1.2.1.130
   Vpi                   =  1
   Vci                   =  121
   AdminStatus           =  Up
   OperStatus            =  Up
   LastChange            =  0 day 0 hour 0 min 0.0 sec
                            (1999/12/23-04:03:39)
   AalType               =  Aal5
   Aal5CpcsTransmitSduS =  1600
   Aal5CpcsReceiveSduSi =  1600
   RowStatus             =  Active
   TransmitTrafficDescr =  1
   OAMState              =  adminDown(0)
   OAMAdminState         =  disabled
   OAMAutoLBState        =  disabled
   OAMManualLBCmd        =  none
   OAMManualLBCmdStatus =  none
   OAMLBInterval         =  5
   OAMLBTimeOut          =  15
   OAMTxAISCells         =  0
   OAMRxRDICells         =  0
   OAMRxAISCells         =  0
   OAMTxRDICells         =  0
   OAMTxLBRequestCells  =  0
   OAMRxLBResponseCells =  0
   OAMRxLBRequestCells  =  0
   OAMTxLBResponseCells =  0
   OAMRxUnsupportedCell =  0
   OAMTxDiscards         =  0
   OAMRxDiscards         =  0
   ```

2. Set the NetModel and the EncapsulationType for the WAN VC. The NetModel must be Cross-Connect. The EncapsulationType must be None.

   ```
   CRAFT> set cmiface [1.2.1.130] netmodel=cross-
   connect encapsulationtype=none
   ```

   Perform a Get on the VC and you will see this table:

   ```
   CRAFT> get cmiface [1.2.1.130]
   Group: cmIfaceTable
   Instance: [1.2.1.130]
   PII                   =  1.2.1.130
   IfIndex               =  1.2.1.130
   Name                  =  ""
   GroupName             =  ""
   AdditionalInfo        =  ""
   NetModel              =  Cross-Connect
   IpAddr                =  0.0.0.0
   NetMask               =  0.0.0.0
   MacAddr               =  ff.ff.ff.ff.ff.ff
   BurnedInMacAddr       =  ff.ff.ff.ff.ff.ff
   FarEndAddr            =  0.0.0.0
   DestPII               =  0.0.0.0
   CMCPCompatible        =  No
   EncapsulationType     =  None
   FwdMode               =  VC-VC-payload
   Pix                   =  274
   ServiceClass          =  None
   ```

3. Set the VC on the DSL port. Use a PII such as 1.4.1.23 and specify a VPI and VCI. They must match the ones on the CPE. Typically, they are 0 and 32. Documentation for CPEs will give defaults, indicate if they can be changed, and show when multiple VCs are possible. The number of the VC must be in the range of 16 to 23.

   ```
   CRAFT> set cmatmvcl [1.4.1.23] vpi=0 vci=32
   adminstatus=up
   ```

   Perform a Get on the VC and you will see this table:

   ```
   CRAFT> get cmatmvcl [1.4.1.23]
   Group: cmAtmVclTable
   Instance: [1.4.1.23]
   PII                   =  1.4.1.23
   Vpi                   =  0
   Vci                   =  32
   AdminStatus           =  Up
   OperStatus            =  Up
   LastChange            =  0 day 0 hour 0 min 0.0 sec
                            (1999/12/23-08:03:37)
   AalType               =  Aal5
   Aal5CpcsTransmitSduS  =  1600
   Aal5CpcsReceiveSduSi  =  1600
   RowStatus             =  Active
   TransmitTrafficDescr  =  1
   OAMState              =  adminDown(0)
   OAMAdminState         =  disabled
   OAMAutoLBState        =  disabled
   OAMManualLBCmd        =  none
   OAMManualLBCmdStatus  =  none
   OAMLBInterval         =  5
   OAMLBTimeOut          =  15
   OAMTxAISCells         =  0
   OAMRxRDICells         =  0
   OAMRxAISCells         =  0
   OAMTxRDICells         =  0
   OAMTxLBRequestCells   =  0
   OAMRxLBResponseCells  =  0
   OAMRxLBRequestCells   =  0
   OAMTxLBResponseCells  =  0
   OAMRxUnsupportedCell  =  0
   OAMTxDiscards         =  0
   OAMRxDiscards         =  0
   ```

4. Set the NetModel and the EncapsulationType for the DSL port. Use the same PII, 1.4.1, that you used in step 3. The VC number is not necessary. Set CMCP to No (the default is Yes).

   ```
   CRAFT> set cmiface [1.4.1] netmodel=cross-connect
   encapsulationtype=atm cmcpcompatible=no
   ```

   Perform a Get on the port and look at the table:

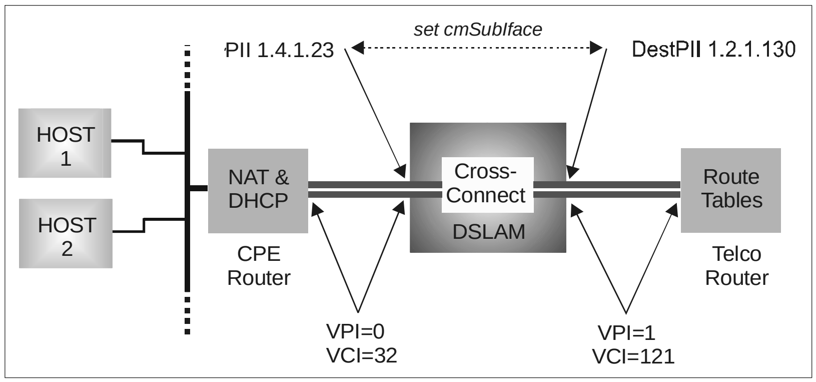
   ```
   CRAFT> get cmiface [1.4.1]
   Group: cmIfaceTable
   Instance: [1.4.1.0]
   PII                   =  1.4.1.0
   IfIndex               =  1.4.1.0
   Name                  =  ""
   GroupName             =  ""
   AdditionalInfo        =  ""
   NetModel              =  Cross-Connect
   IpAddr                =  0.0.0.0
   NetMask               =  0.0.0.0
   MacAddr               =  ff.ff.ff.ff.ff.ff
   BurnedInMacAddr       =  0.0.0.0.0.0
   FarEndAddr            =  0.0.0.0
   DestPII               =  0.0.0.0
   CMCPCompatible        =  No
   EncapsulationType     =  atm
   FwdMode               =  PER-VC
   Pix                   =  3
   ServiceClass          =  D
   ```

5. Set the VC on the DSL port so that it points toward the VC on the WAN port. All data from the VC on the DSL port will automatically go to the VC on the WAN port, and vice versa. Use the DestPII setting to accomplish the mapping.

```
CRAFT> set cmsubiface [1.4.1.23] destpii=1.2.1.130
```

Perform a Get on the VC and you will see this table:

```
CRAFT> get cmsubiface [1.4.1.23]
Group: cmSubIfaceTable
Instance: [1.4.1.23]
PII                  =  1.4.1.23
DestPII              =  1.2.1.130
Name                 =  ""
RowStatus            =  Active
Priority             =  Low
```



Cross-Connect Netmodel—Destination PII

IP Netmodel

In the IP netmodel, the end points of the VCs must be manually configured for each of the stages on the link, but this time, each of the interfaces requires an IP address. (In the diagram below, all addresses are private, although they could be public.)

The VC going from the Telco router to the WAN port on the CE150 must have an IP address at each end. Its endpoints must be on different subnets. The WAN VC and the DSL port also must have IP addresses. They also must be on different subnets. Finally, the DSL port and its downstream hosts must have IP addresses, but they must be on the same subnet.

Since the CPE is in bridge mode, it doesn't require an IP address. The DSL port, not the upstream CPE port, is the gateway for hosts on the LAN. But, since the CE150 in the IP netmodel is in router mode, it sends packets coming upstream from its DSL ports to its WAN VC based on destination addresses in the packets. It sends packets coming downstream from its WAN VC to the proper DSL ports based on destination addresses.

For each of the DSL ports with G.dmt and G.lite CPEs connected to them, part of the configuring process must be repeated. New DSL VCs must be created on each DSL port.

In the following diagram, only one DSL and one WAN port appear. In the procedure on the following pages, first a WAN VC with an IP address of 192.168.20.2 will be created; then a DSL VC with an IP address of 10.10.1.1. Finally, the default IP route on the CE150 will be set. Configuration of addresses on the Telco router and downstream hosts will not be shown.

1. Set the VC on the WAN port. The VPI and VCI on the WAN VC must match those on the Telco router. Set AdminStatus to Up (the default is Down).

   ```
   CRAFT> set cmatmvcl [1.2.1.80] vpi=1 vci=122
   adminstatus=up
   ```

   Perform a Get on the VC and you will see this table:

   ```
   CRAFT> get cmatmvcl [1.2.1.80]
   Group: cmAtmVclTable
   Instance: [1.2.1.80]
   PII                  =  1.2.1.80
   Vpi                  =  1
   Vci                  =  122
   AdminStatus          =  Up
   OperStatus           =  Up
   LastChange           =  0 day 0 hour 0 min 0.0 sec
                           (2000)04/14-14:41:37)
   AalType              =  Aal5
   Aal5CpcsTransmitSduS =  1600
   Aal5CpcsReceiveSduSi =  1600
   RowStatus            =  Active
   TransmitTrafficDescr =  1
   OAMState             =  adminDown(0)
   OAMAdminState        =  disabled
   OAMAutoLBState       =  disabled
   OAMManualLBCmd       =  none
   OAMManualLBCmdStatus =  none
   OAMLBInterval        =  5
   OAMLBTimeOut         =  15
   OAMTxAISCells        =  0
   OAMRxRDICells        =  0
   OAMRxAISCells        =  0
   OAMTxRDICells        =  0
   OAMTxLBRequestCells  =  0
   OAMRxLBResponseCells =  0
   OAMRxLBRequestCells  =  0
   OAMTxLBResponseCells =  0
   OAMRxUnsupportedCell =  0
   OAMTxDiscards        =  0
   OAMRxDiscards        =  0
   ```

2. Set NetModel, IPAddr, NetMask, and EncapsulationType for the WAN port and the FarEndAddr. The NetModel must be IP. The EncapsulationType must be rfc1483.

   ```
   CRAFT> set cmiface [1.2.1.80] netmodel=ip
   ipaddr=192.168.20.2 netmask=255.255.255.255
   farendaddr=192.168.20.1 encapsulationtype=rfc1483
   ```

   Perform a Get on the VC and you will see this table:

   ```
   CRAFT> get cmiface [1.2.1.80]
   Group: cmIfaceTable
   Instance: [1.2.1.80]
   PII                  =  1.2.1.80
   IfIndex              =  1.2.1.80
   Name                 =  ""
   GroupName            =  ""
   AdditionalInfo       =  ""
   NetModel             =  IP
   IpAddr               =  192.168.20.2
   NetMask              =  255.255.255.255
   MacAddr              =  ff.ff.ff.ff.ff.ff
   BurnedInMacAddr      =  ff.ff.ff.ff.ff.ff
   FarEndAddr           =  192.168.10.1
   DestPII              =  0.0.0.0
   CMCPCompatible       =  No
   EncapsulationType    =  rfc1483
   FwdMode              =  Full-IP
   Pix                  =  28
   ServiceClass         =  None
   ```

3. Set the VC on the DSL port. Use a PII such as 1.4.12.528 and specify a VPI and VCI. They must match the ones on the CPE. Typically, they are 0 and 32. The number of the VC itself must be 528. It cannot be any other number for rfc1483.

```
CRAFT> set cmatmvcl [1.4.12.528] vpi=0 vci=32
adminstatus=up
```

Perform a Get on the VC and you will see this table

```
CRAFT> get cmatmvcl [1.4.12.528]
Group: cmAtmVclTable
Instance: [1.4.12.528]
PII                    =  1.4.12.528
Vpi                    =  0
Vci                    =  32
AdminStatus            =  Up
OperStatus             =  Up
LastChange             =  0 day 0 hour 0 min 36.0 sec
                          (2000/04/14-14:42:13)
AalType                =  Aal5
Aal5CpcsTransmitSduS   =  1600
Aal5CpcsReceiveSduSi   =  1600
RowStatus              =  Active
TransmitTrafficDescr   =  0
```

4. Set the NetModel and the EncapsulationType for the DSL port. Use the same PII, 1.4.12, that you used in step 3. The VC number is not necessary. Set CMCPCompatible to No (the default is Yes).

```
CRAFT> set cmiface [1.4.12] netmodel=ip
encapsulationtype=rfc1483 cmcpcompatible=no
```

Perform a Get on the port and look at the table:

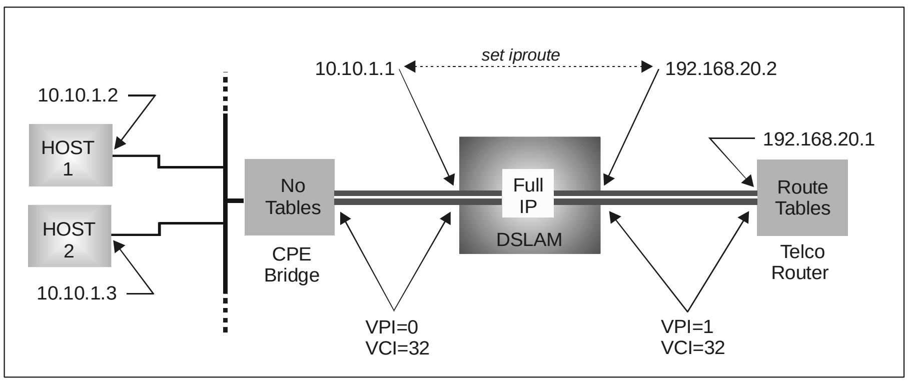
```
CRAFT> get cmiface [1.4.12]
Group: cmIfaceTable
Instance: [1.4.12.0]
PII                    =  1.4.12.0
IfIndex                =  1.4.12.
Name                   =  ""
GroupName              =  ""
AdditionalInfo         =  "
NetModel               =  IP
IpAddr                 =  10.10.1.1
NetMask                =  255.255.255.0
MacAddr                =  ff.ff.ff.ff.ff.ff
BurnedInMacAddr        =  0.0.0.0.0.0
FarEndAddr             =  0.0.0.0
DestPII                =  0.0.0.0
CMCPCompatible         =  No
EncapsulationType      =  rfc1483
FwdMode                =  Full-IP
Pix                    =  14
ServiceClass           =  D
```

5. Perform a set for the default IP route leading from the DSL ports to the WAN VC and, alternatively, from the VC to the DSL ports. The IP netmodel should work without a default IP route, however, since the CE150 acts as a router for the traffic going between the DSL ports (VCs) and the WAN ports (VCs).

```
CRAFT> set iproute [0.0.0.0] nexthop=192.168.20.1
type=direct
```

Perform a Get and look at the table:

```
CRAFT> get iproute [0.0.0.0]
Group: ipRouteTable
Instance: [0.0.0.0]
Dest                 =  0.0.0.0
IfIndex              =  1.2.1.80
Metric1              =  1
Metric2              =  -1
Metric3              =  -1
Metric4              =  -1
NextHop              =  192.168.20.2
Type                 =  Direct
Proto                =  netmgmt
Age                  =  755624
Mask                 =  0.0.0.0
Metric5              =  -1
Info                 =  0.0
```



IP Netmodel—Default Route

Chapter 8
Troubleshooting

This chapter describes features of the CE150 that can aid in evaluating system performance, and in diagnosing and recovering from trouble.

CE150 Diagnostic Features

The CE150 includes features to help you to diagnose problems in case something should go wrong. Should a malfunction occur, these features can help you to pinpoint the device causing the problem, and indicate possible avenues for recovery.

Diagnostic features include:

- Visual indicators (front panel LEDs)
- Traps and alarms
- Loopbacks and Loop Quality Tests
- Built-in performance monitors
- Diagnostic port (assisted by Copper Mountain Technical Support)

Front Panel LED Indicators

Module Status Indicators

Circuit modules for the CE150 are equipped with front-panel LEDs that show module functionality at a glance. The module status LEDs on the AC and DC power modules, DSL line modules, and Buffer Control Module indicate the general status of the circuit pack whenever power is applied.

Under normal circumstances, the module status indicator LEDs should be on and green. The Buffer Control Module status LED provides additional information as discussed below.

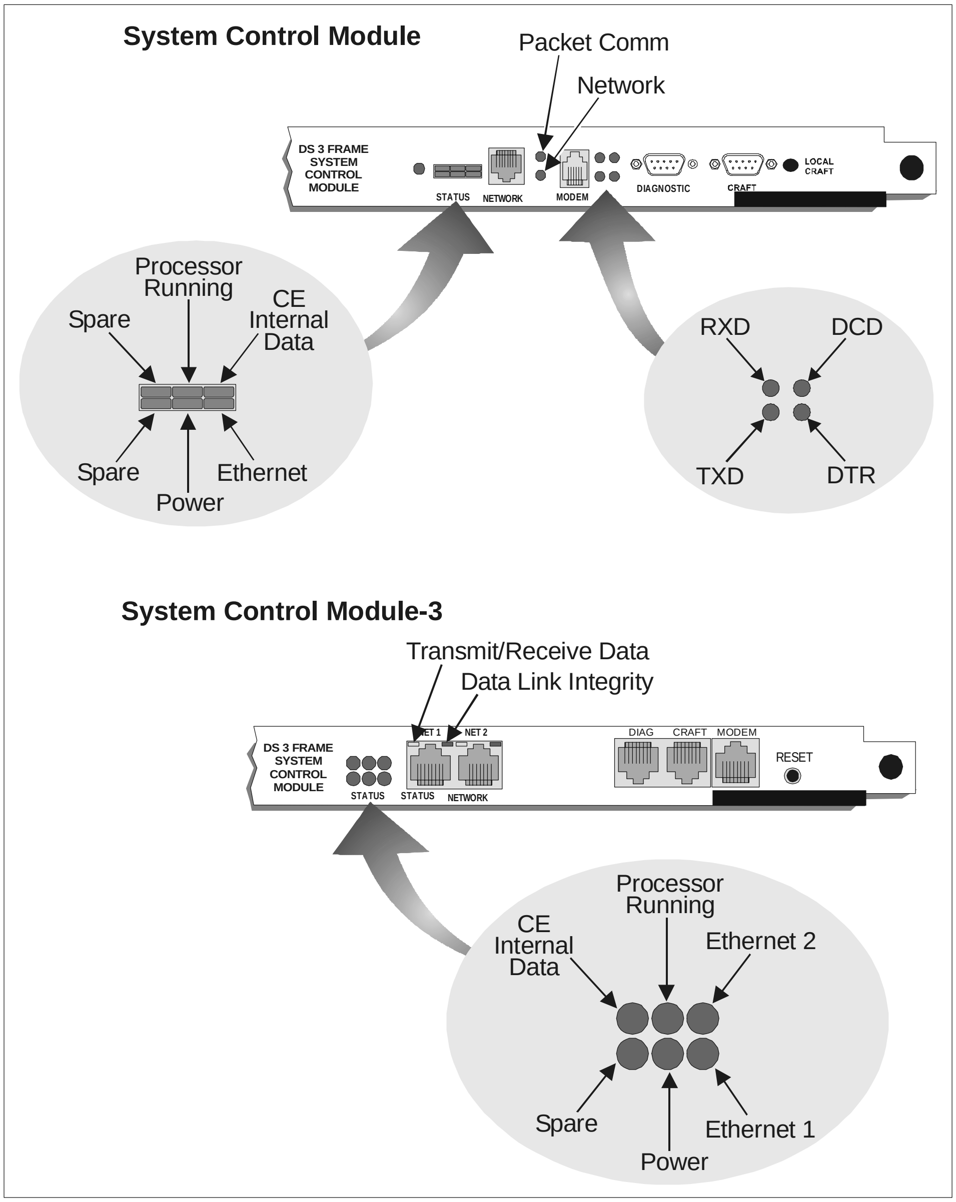
Buffer Control Module Status

During normal operation (startup sequence completed), the Buffer Control Module should be green and flickering to indicate data flow. In case of trouble, however, Buffer Control Modules of Version 2.1.63 and later could also display the following:

- Alternating *Red*/*Amber*—Abnormal operation. At least one invalid descriptor has been received from the System Control Module, the affected DSL ports have been deactivated, unaffected DSL ports will continue to operate normally.
- Continuous cycle of *Red*/*Amber*/Off/*Green*—The Buffer Control Boot PROM cannot properly initialize the module. Not usually indicative of a failure in the Buffer Control Module, but more likely in the PCI bus or other PCI device. Restarting the System Control Module may clear this condition; cycling power to the CE150 should always clear it.
- *Red*—Buffer Control Module is in latched reset because the CPU has failed to start, or the SRAM test failed. A full CE150 power cycle is required to clear this state.

System Control Module Status

The System Control Module and System Control Module-3 have six LEDs beside the Ethernet connector. When power is applied, the Power LED immediately turns green, and the Processor Running indicator turns green within a second or two.



System Control Module LEDs

If the System Control Module is connected to an Ethernet network, the module will synchronize and negotiate its speed with the network. The Ethernet LED turns green if the network is a 10BASE-T (10 Mbps) system, and red if it is a 100BASE-T (100 Mbps) system.

N O T E

A green-lit Ethernet LED is not a signal that the CE150 is connected to or communicating with a live Ethernet network; the LED turns green by default, whether or not the port is actually connected.

The CE Internal Data LED indicates data activity internal to the CE150 itself. The state of this LED does not correspond to the status of the system. At any time, the Internal Data LED may be off, blinking, pulsing, or solidly green.

On the System Control Module panel, the Network and Packet Comm LEDs beside the modem port indicate data traffic. The Network indicator turns or flickers green when Ethernet data packets are transmitted by the CE150. The Packet Comm indicator turns or flickers green when Ethernet data packets are received by the CE150.

On the System Control Module-3 panel, the Ethernet port has two LEDs: a single Transmit/Receive Data LED and a Data Link Integrity LED. The Transmit/Receive Data LED indicates data traffic. It turns or flickers yellow when data is received or transmitted on the Ethernet line. The Data Link Integrity LED turns green when the integrity of the Ethernet link for 10BASE-T or 100BASE-TX is acceptable.

Modem Indicators

On the System Control Module, the four LEDs for the modem port indicate two things: handshaking between the sending and receiving terminals; and the actual sending and receiving of data. The LEDs for Data Communications Detect (DCD] and Data Terminal Ready [DTR] start flickering during handshaking and then turn a steady green when sending or receiving data. The LEDs for Received [RXD} and Transmitted (TXD) turn green for the duration of receiving or transmitted data.

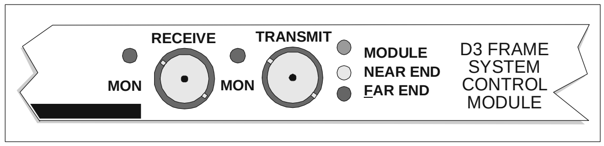
The modem is set to automatically answer on the second ring. However, it will take longer to answer when it is trying to reconnect after a disconnection—it cannot answer on the second ring because it is still busy disconnecting from the previous session.

Craft Port Indicators

The single LEDs for the Local Craft port will turn yellow as soon as a cable has been attached and the System Control Module detects a device on the other end.

DS3 WAN Module Status

The DS3 WAN module (ATM or Frame Relay) has three LED indicators on its front panel, to the right of the Transmit BNC connector. At power-up, all three LEDs briefly illuminate as a functional test.



DS3 WAN Module LEDs

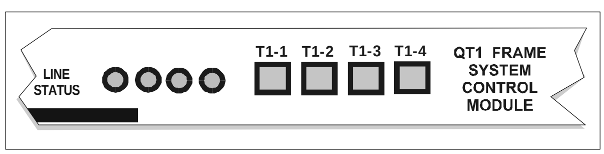
If the DS3 WAN module is connected to a DS3 ATM or Frame Relay facility through a suitable access device, the green Module LED illuminates, and the red Near End and yellow Far End LEDs are off.

If there is a local failure, such as loss of signal or frame errors at the CE150 or within the DS3 module, the NEAR END LED turns red to indicate the failure.

If a failure occurs on the link or with upstream equipment, or when diagnostic testing is in progress, the FAR END LED turns yellow. When a software download is in progress, the FAR END LED blinks.

Quad T1 Frame WAN Module Indicators

The DS1 WAN module has four LED indicators that indicate the condition of the T1 line or the type of traffic on the T1 line, including alarm signals and normal operating signals.



DS1 WAN Module LEDs

The LEDs are to the left of the connectors, in the same order as the connectors (for example, the left LED is for port T1-1, and the right LED is for port T1-4). Each LED has three colors that indicate different states:

- *Green*—Normal operation. The line has been synchronized with the network and shows no local or far-end alarms.

- *Yellow*—A far-end failure condition on the link or with upstream equipment. The LED stays yellow until the condition is cleared.
- *Red*—A near-end failure condition, such as a loss of signal or frame errors at the CE150, or within the Quad T1 Frame Module itself. The LED stays red until the condition is cleared.

SDSL and ADSL (G.dmt or G.lite) Module Port Status

All SDSL and ADSL (G.dmt or G.lite) modules have a cluster of 24 LED indicators, each of which corresponds to one of its DSL ports.

If a port has been correctly wired and configured, the CPE at the far end of the DSL link will train on the port and the associated LED will turn green. If the indicator is off, or if it continually alternates between off and on, there is a fault with the port, link, or CPE.

T1 DSL Line Module Port Status

The T1 DSL line module has a cluster of 12 LED indicators, each of which corresponds to one of its DSL ports. Each LED has three colors that indicate the following states:

- *Off*—Reset state, power down state, or administratively down (Admin=Down).
- *Green*—Normal operation, synchronized with the network; no local or far-end alarms; administratively up (Admin=Up).
- *Yellow*—Far-end failure with no near-end failure; administratively up (Admin=Up).
- *Red*—Near-end failure, such as loss of signal, loss of frame, or receiving AIS; with or without a far-end failure; administratively up (Admin=Up).

Events and Alarms

The CE150 is designed to be in continuous operation, 24 hours a day. During operation, various events will occur: operators will log in and out; configuration files will be modified and saved; the status of DSL or network links will change; and occasional failures or other operational anomalies may occur.

Events can be sent by any resource in the system. They are logged in memory (locally on the CE150 system), and are also translated into SNMP Trap PDUs to be sent to external SNMP managers. Here are the three groups of events on the CopperEdge:

- *Alarms*—Events that indicate the occurrence of an alarm condition; alarms will affect the status of the alarm panel; in general an alarm will remain in effect until cleared (either automatically or manually).
- *Alarm Clearing Notifications*—Events that clear a particular alarm (if one exists); if no corresponding alarm condition exists for such an event to clear, it is treated as a normal notification.
- *Event Notifications*—Events, such as warnings and informational messages, that neither trigger nor clear an alarm.

For a list of Alarms, Alarm Clearing Notifications, and Event Notifications that may occur on your CopperEdge system, see Appendix C.

Event Contents

Each event contains a set of data that is common to all events, plus a set of data (possibly empty) that is specific to the particular event. The common data is used to identify the type of event, the resource reporting the event, and the severity of the event. The event-specific data is included to provide more details about the particular event. The common data contained in every event consists of the following objects from the cmTrapEventTable: SeqNum, Type, ObjectClass, ClassId, Severity, and TimeTag.

Resource Identification

The resource that reported the event can be uniquely identified using two fields: ObjectClass and ClassId. Each resource in the system belongs to a specific object class (such as Board or Port). The specific resource within the class can be identified using the ClassID, which is the PII of the resource.

The class identifies the type of resource reporting the event and the ClassID identifies the specific resource in that class. For example, in a BoardDown event, the ObjectClass might be SystemControlModule and the ClassId would be 1.2.0.0. This event indicates the SCM in the second slot has gone down.

Object Classes

The following object classes are used to represent all of the resources in the CE150 system.

System Classes

These classes apply to the system as a whole.

- System
- Operator

Shelf Classes

These classes identify the different shelves in a system.

- Shelf
- CE200Shelf
- CE150Shelf

Board Classes

These classes identify the different boards that can be installed in a system.

- Board
- SystemControlModule
- CopperBayModule
- SDSL10xModule-1
- SDSL10xModule
- SDSL30xModule
- SDSL30xModule-24
- IDSLModule-24
- BufferControlModule
- V.35-WAN
- DS3FR-WAN

- DS3ATM-WAN
- QuadT1-WAN
- G.liteModule-24
- T1Module-12
- SystemControlModule3
- ADSLMultiModeModule-24A

Port Classes

These classes identify the different physical interfaces in a system.

- Port
- SDSLPort
- EthernetPort
- V.35Port
- RS232Port
- DS3FRPort
- IDSLPort
- DS3ATMPort
- T1Port-WAN
- T1Port
- G.litePort
- G.dmtPort
- T1Port-LC

Link Classes

These classes identify a logical data connection that runs over a physical interface.

- Link
- LCPortLink
- EthernetLink
- FrameRelayLink
- FrameRelayPVC
- ATMLink
- InterShelfTrunk

CPE Classes

These classes identify the different CPEs connected to the system.

- CR
- CR201-10x
- CR201-30x
- CR201-SDSL
- CR201-IDSL
- CPE-SDSL

- CPE-IDSL
- Netopia-SDSL
- CPE-T1

Support Classes

These classes identify resources that are supporting entities in the system.

- Support
- PowerModule
- FanModule

Logs

All events that occur on the CE150 are captured in three locations:

- The Event Log contains all monitored events, whether a major alarm or a notification.
- The Alarm Table contains all alarms.
- The Audit Log contains a record of attribute changes: the exact command entered to effect each change, and the IP address and name of the operator originating the command.

Event Log

As events occur, the information is logged, and can be viewed in real time or captured to a file. The CE150 maintains a running record of the most recent 1,000 events since the system was last reset.

To view the Event Log in real time, connect to the system either directly or through Telnet and enter the command, **elog**. In this mode, events are displayed on the screen as they occur. You can also capture the information by logging your Telnet session and saving it for later analysis. To exit the elog function, press **Esc**.

```
CRAFT> elog
2000/12/03-09:36:32     LoginFail
   cmActiveSessionTable  IpAddress         = 10.254.8.202
   cmTrapEventTable      Text              = admin, FTP
   cmTrapEventTable      Type              = LoginFailed
   cmTrapEventTable      ObjectClass       = Operator
   cmTrapEventTable      ClassId           = 1.2.0.0
   cmTrapEventTable      SeqNum            = 67
   cmTrapEventTable      TimeTag           = 2000/12/03-
                                             09:36:32
   cmTrapEventTable      Severity         = Information
2000/12/03-09:37:42      LoginSucceed
   cmActiveSessionTable  IpAddress         = 10.64.20.190
   cmTrapEventTable      Text              = ce200, Telnet
   cmTrapEventTable      Type              = LoginSucceeded
   cmTrapEventTable      ObjectClass       = Operator
   cmTrapEventTable      ClassId           = 1.2.0.0
   cmTrapEventTable      SeqNum            = 68
```

```
cmTrapEventTable      TimeTag            = 2000/12/03-
                                           09:37:42
cmTrapEventTable      Severity           = Information
```

Event log information can also be viewed through the cmTrapEvent group (described in the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual).

```
CRAFT> geta cmtrapevent
SeqNum           Type               ObjectClass         ClassId
ProbableCause    Severity           TimeTag             Text


Instance: [65]
65               LoginFailed        Operator            1.2.0.0
0               Information       2000/12/03-05:36:28 10.254.8.202, admin


Instance: [66]
66               LoginFailed        Operator            1.2.0.0
0               Information       2000/12/03-07:36:30 10.254.8.202, admin


Instance: [67]
67               LoginFailed        Operator            1.2.0.0
0               Information       2000/12/03-09:36:32 10.254.8.202, admin


Instance: [68]
68               LoginSucceeded     Operator            1.2.0.0
0               Information       2000/12/03-09:37:42 10.64.20.190, ce200
```

Or,

```
CRAFT> get cmtrapevent [67]
Group: cmTrapEventTable
Instance: [67]
SeqNum                  =  67
Type                    =  LoginFailed
ObjectClass             =  Operator
ClassId                 =  1.2.0.0
ProbableCause           =  0
Severity                =  Information
TimeTag                 =  2000/12/03-09:36:32
Text                    =  10.254.8.202, admin, FTP
```

Alarm Log

Alarm messages are written to the event log, and also to a separate alarm table that is updated every time the alarm status changes. Once an alarm condition is cleared, it is dropped from the table.

Like the Event Log, two different methods are available for viewing information about current alarms. To view the Alarm Table in real time, connect to the system either directly or through Telnet and enter the command, **`alarms`**. A list of all current (uncleared) alarms will be displayed. To exit the Alarm Table, press **Esc**.

```
CRAFT> alarms
*********
PortMisprovisioned  (2000/12/02-11:49:53) id=25 (MINOR)(IDSLPort:1.11.4.0)
PortMisprovisioned  (2000/12/02-11:49:55) id=27 (MINOR)(IDSLPort:1.11.14.0)
PortMisprovisioned  (2000/12/02-11:50:29) id=33 (MINOR)(SDSLPort:1.10.14.0)
PortMisprovisioned  (2000/12/02-11:50:29) id=33 (MINOR)(SDSLPort:1.10.14.0)
```

The second method is to view Alarm table information through the cmTrapAlarm group (described in the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual).

```
CRAFT> geta cmtrapalarm
SeqNum          Type                ObjectClass       ClassId
ProbableCause   Severity            TimeTag              Text


Instance: [25]
25              PortMisprovisioned  IDSLPort          1.11.4.0
0              MinorAlarm         2000/12/02-11:49:53 Operation state of


Instance: [27]
27              PortMisprovisioned  IDSLPort          1.11.14.0
0              MinorAlarm         2000/12/02-11:49:55 Operation state of


Instance: [33]
33              PortMisprovisioned  SDSLPort          1.10.14.0
0              MinorAlarm         2000/12/02-11:50:29 Operation state of
```

Or,

```
CRAFT> get cmtrapalarm [33]
Group: cmTrapAlarmTable
Instance: [33]
SeqNum              =  33
Type                =  PortMisprovisioned
ObjectClass         =  SDSLPort
ClassId             =  1.10.14.0
ProbableCause       =  0
Severity            =  MinorAlarm
TimeTag             =  2000/12/02-11:50:29
Text                =  Operation state of the port is up,
                       but the port is not configured
```

Audit Log

When isolating the cause of a problem, it can be helpful to know what might have triggered it. On systems with an SCM-2 or SCM-3, the CE150 software includes uploadable logfiles containing a record of the most recent attribute changes (up to 1,000 events) on this CE150. Attribute changes are generally changes that result from a Set command.

With the time-stamped AttributeChange known, you can logically infer the precipitating command. The Audit Log also lists the name and IP address of the operator initiating the change, as well as the command context (such as Telnet or SNMP) and the privilege level of the operator. The addition of this information invests the Audit Log with an implicit role in maintaining system security.

The full audit log information base consists of four files stored in the Logs directory on the CE150 flash system (the P: (IDE) drive) of the SCM-2 and SCM-3:

- *audit.txt*—The audit log that is currently filling; contains the most recent AttibuteChange events.
- *audit_count.txt*—This file contains a running count of the events in the *audit.txt* file. When this number reaches 500, new *audit.txt* and *audit_count.txt* files are created and the old files are renamed.
- *audit.bak*—The previous *audit.txt* file, now filled with the 500 AttributeChange events immediately preceding those in the current *audit.txt* file.
- *audit_count.bak*—The previous *audit_count.txt* file, now indicating the total number of records (500) in the *audit.bak* file. When the next cycle is completed and the current *.txt* files are filled and renamed, the current *.bak* files are overwritten.

The audit log files are accessible only through FTP, and only by an operator with Security level privilege. Once uploaded, the raw files have the format of one record per line, with fields delimited by a vertical bar (|), but this can be easily re-formatted using a spread-sheet program.

Traps

In the CE150, every internal event generates an SNMP *trap*. A trap is a record of the event that is captured and sent to the configured trap destinations for the system; it contains the same information as the event. For a listing of all currently supported traps associated with each event, see Appendix C.

There is no one-to-one mapping between events and traps. Because of the way alarms are cleared internally, one or more events may translate into the same trap. In these cases, one or more data fields can be used to distinguish the traps. For example, both the DLCIStateDisabled and DLCIStateEnabled events result in a frDLCIStatusChange trap being generated.

To view a chronological list of the 1,000 most recent traps (events) you can use the elog or cmTrapEvent group as described above. The same information is also available sorted by the type of trap or event. To access this list, use the cmTrapType group as described in the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual.

Traps can be sent to as many as ten different SNMP managers. The list of those designated to receive traps is contained in the cmTrapDestination table in the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual.

Alarms

Alarms constitute one subset of events. When generated, an alarm message contains the following information:

- Name of the alarm (such as FanFailure)
- Source (module, port, link)
- Severity (Critical, Major, Minor)
- Time the event occurred
- Other event-specific information

Alarm Severity

Alarms are divided into three levels of severity:

- *Critical*—A service-affecting condition that requires immediate corrective action. Used when the entity is totally out of service and its capability must be restored.
- *Major*—A service-affecting condition that requires urgent corrective action. Used when there is a severe degradation of the entity's capability, which must be restored.
- *Minor*—An event or condition that does not affect service, but corrective action should be taken to prevent a more serious fault.

Alarm Clearing

The phrase “clear an alarm” describes an occurrence (a CE150-generated event or manual action) that causes visual and audible alarm indications to disappear (usually but not always corresponding to correction of the underlying problem).

Some alarms (such as BoardDown) are cleared by the CE150 when a corresponding event (in this case, BoardUp) occurs. Other alarms (such as RoleChanged) have no such corresponding event. In the latter case, you must manually clear the alarm by setting the cmAlarmTable as discussed in the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual.

Manually clearing an alarm in the CE150 does not imply fixing the problem. Clearing the alarm merely causes the alarm unit to no longer display alarm indications.

You can manually clear alarms that support automatic clearing. For example, you would do this if a line card is being removed from a system with no immediate plans to replace it. The alarm that would result from the subsequent BoardDown event could be suppressed by manually clearing the alarm. However, we recommend that you do not manually clear such alarms very often, because information about the status of your system will be discarded in the process.

When an alarm is cleared, a *Notification* message is displayed showing the new operational condition. Not all Notification messages are alarm-clearing; some are simply informational or advisory. A *Warning* is essentially a Notification, but with the added dimension of calling the operator’s attention to a situation that may include some vulnerability or future risk.

Example Alarms and Events

Elog Display

This example shows how the events are seen using the elog command.

Here is a link down alarm indicating that the Frame Relay Link on port [1.2.1] has gone down:

```
1999/08/18-17:38:38     LinkDown
   ifTable            Index        = 1.2.1.0
   cmNameTable        IfName       = ""
   cmNameTable        PII          = 1.2.1.0
   cmGroupTable       Name         = ""
   cmTrapEventTable   Type         = LinkDown
   cmTrapEventTable   ObjectClass  = FrameRelayLink
   cmTrapEventTable   ClassId      = 1.2.1.0
   cmTrapEventTable   SeqNum       = 56
   cmTrapEventTable   TimeTag      = 1999/08/18-
                                     17:38:38
   cmTrapEventTable   Severity     = MinorAlarm
```

Here is a link up clearing notification for the Frame Relay Link on port [1.2.1]:

```
1999/08/18-19:21:12     LinkUp
   ifTable            Index        = 1.2.1.0
   cmNameTable        IfName       = ""
   cmNameTable        PII          = 1.2.1.0
   cmGroupTable       Name         = ""
   cmTrapEventTable   Type         = LinkUp
   cmTrapEventTable   ObjectClass  = FrameRelayLink
   cmTrapEventTable   ClassId      = 1.2.1.0
   cmTrapEventTable   SeqNum       = 249
   cmTrapEventTable   TimeTag      = 1999/08/18-
                                     19:21:12
   cmTrapEventTable   Severity     = Information
```

cmTrapEvent Display

This example shows how the events are displayed using the cmTrapEventLogTable. You can view the events using the cmTrapEvent group with the following command:

```
CRAFT> get cmtrapevent [sequence_number]
```

The table is sorted by sequence number. In the cmTrapEvent output, the event specific fields are collapsed into a single field called Text. The contents of each event specific field is listed in order and separated by commas.

Here is the output of the LinkDown event for a Frame Relay Link on port 1.2.1:

```
CRAFT> get cmtrapevent [56]
Group: cmTrapEventTable
Instance: [56]
Type                 =  LinkDown
SeqNum               =  56
ObjectClass          =  FrameRelayLink
ClassId              =  1.2.1.0
ProbableCause        =  0
Severity             =  MinorAlarm
TimeTag              =  1999/08/18-17:38:38
Text                 =  1.2.1.0, "", 1.2.1.0, ""
```

Here is the output of the LinkUp event for the Frame Relay Link on port 1.2.1:

```
CRAFT> get cmtrapevent [249]
Instance: []
Type                 =  LinkUp
SeqNum               =  249
ObjectClass          =  FrameRelayLink
ClassId              =  1.2.1.0
ProbableCause        =  0
Severity             =  Information
TimeTag              =  1999/08/16-15:58:08
Text                 =  1.6.4.0, "", 1.6.4.0, ""
```

Both of these events have four event-specific fields (the form is *GroupName*::*ObjectName*):

ifTable::Index
cmNameTable::IfName
cmNameTable::PII
cmGroupTable::Name

Save Configuration Failure

If the system configuration is not successfully saved when you log out of the CE150, the following message is displayed and the logout process is stopped:

```
CE configuration has been changed.
Save the configuration?(Y / y)>y
.....
CE configuration save failed.
```

To see information about the failure, look for the ConfigWrite-Failed event in the event log (it will be at or near the end of the event log).

```
CRAFT> elog
2000/10/03-14:37:03 ConfigWriteFailed      .
  cmSystem           ConfigFileName       = config.tgz
  cmTrapEventTable   Type                = ConfigWriteFailed
  cmTrapEventTable   ObjectClass          = System
  cmTrapEventTable   ClassId              = 1.0.0.0
  cmTrapEventTable   SeqNum               = 11
  cmTrapEventTable   TimeTag              = 2000/10/03-
                                            14:37:03
  cmTrapEventTable   Severity             = Information
```

To see the entire description of the event, find the instance number the ConfigWriteFailed event in the cmTrapEvent group:

```
CRAFT> geta cmtrapevent
SeqNum       Type              ObjectClass        ClassId
ProbableCause Severity         TimeTag            Text

Instance: [11]
11           ConfigWriteFailed  System             1.0.0.0
0           Information       2000/10/02-10:22:23 Unable to
```

To view the full description in the cmTrapEvent group for that instance:

```
CRAFT> geta cmtrapevent [11]
Group: cmTrapEventTable
Instance: [11]
SeqNum               =  11
Type                 =  ConfigWriteFailed
ObjectClass          =  System
ClassId              =  1.0.0.0
ProbableCause        =  0
Severity             =  Information
TimeTag              =  2000/10/02-10:22:23
Text                 =  Unable to open the file
```

See Appendix C, *Events and Alarms*, for more information about the ConfigWriteFailed event.

Terminate an Operator Session

An operator may forget to log out and a script may still be running, preventing the CE150 from automatically logging the session out after a time-out expires. Using the cmActiveSession group, you can terminate any Craft, FTP, or SNMP sessions, whether they were opened locally or through the Radius authentication server. But to terminate a session, an operator must have security level privileges.

First, list all active sessions to find the instance of the one you want to terminate:

```
CRAFT> geta cmactivesession
SessionID          OperatorName       Context            Privilege
IpAddress          StartTime          IdleTimeout        RowStatus

Instance: [1982]
1982               private            SNMP               Provision
10.64.20.162       18 day 18 hour 33 m 900               Active

Instance: [1983]
1983               ce200              Telnet             Security
10.64.20.249       18 day 18 hour 36 m 900               Active

Instance: [1984]
1984               public             SNMP               View
10.64.40.66        18 day 18 hour 36 m 900               Active

Instance: [1985]
1983               ce200              Telnet             Security
10.64.20.234       18 day 18 hour 36 m 900               Active
```

You can optionally view the information for that session only:

```
CRAFT> get cmactivesession [1983]
Group: cmActiveSessionTable
Instance: [1983]
SessionID          =  1983
OperatorName       =  admin
Context            =  Telnet
Privilege          =  Security
IpAddress          =  10.64.20.249
StartTime          =  18 day 18 hour 36 min 27.0 sec
                      (2000/11/18-09:19:39)
IdleTimeout        =  900
RowStatus          =  Active
```

To terminate the active session, delete the row from the table:

```
CRAFT> set cmactivesession [1983] rowstatus=destroy
Set successful
```

Any application running in the specified active session is halted, the session is terminated, and this event is listed in the Event log as a termination of the session. The following message is displayed on the operator's screen:

```
2000/11/18-09:36:30  USER LOGGED OUT
```

Repeat the **geta cmactivesession** command to check that the terminated session is no longer in the table.

The cmActiveSession table will not let you accidentally terminate your own session. If you enter your session's instance, the following message will be displayed:

```
CRAFT> set cmactivesession [1985] rowstatus=destroy
Delete failed
Group: cmActiveSessionTable
```

DHCP Problems

The proper functioning of the DHCP feature depends on the correct configuration of the CE150 as well as of the upstream router and DHCP server.

If the CPE or premise LAN host does not receive the expected response for a DHCP request that it originated, follow these diagnostic steps.

In general:

1. Check the cmDHCPTable to see if an entry exists for DSL PII.
2. If it exists, verify that the Function object is set correctly: DHCPRespond, DHCPRelay, or DHCPForward.

For DHCP Relaying:

1. Verify that the upstream router is able to perform the DHCP Relay or DHCP Server function.
2. If the netmodel is IP, verify that the DHCP Server's IP address configured in the cmDHCPTable is reachable from the CE150.

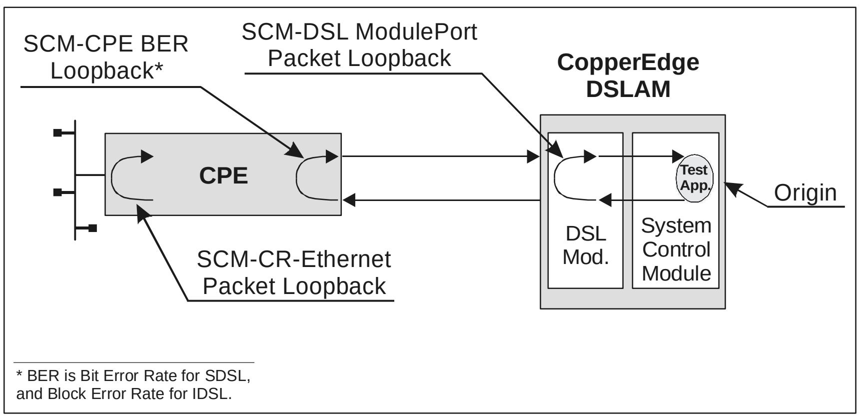
For DHCP Relay and Forwarding:

Check the cmDHCPTable and verify that the CircuitID object is correct. This is used by the upstream DHCP server for generating the appropriate DHCP responses.

Loopbacks

The CE150 can be configured for a number of loopback tests useful not only in isolating failures but also in evaluating line performance over various segments of the end-to-end circuit. In the current CE150 product release, loopbacks are available for test and evaluation of the following: SDSL ports; DS3 ATM and Frame Relay ports; and DS1 ports.

Three DSL loopbacks (between the CE150 main processor and the subscriber CPE) are controlled through the cmLoop group. All of the loopback tests are packet-oriented, except the SCM-CPE BER loopback, which is a bit-error rate test. The cmLoopHistory group provides information on the ten most recent tests run through the cmLoop group. For information on configuring cmLoop and cmLoopHistory, see the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual. The following diagram illustrates the DSL loopback tests supported on the CE150.

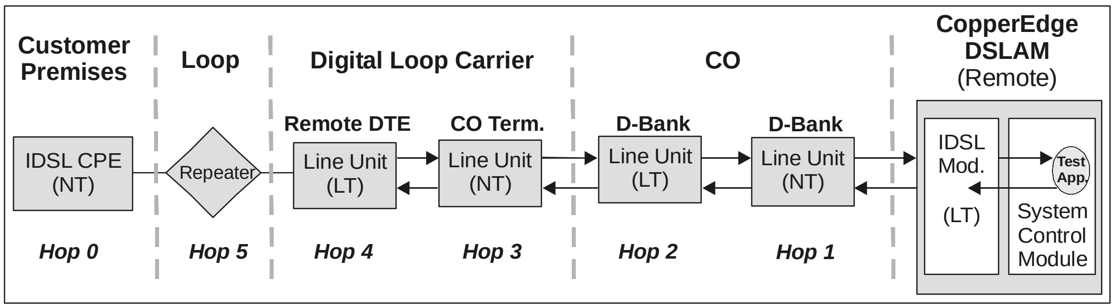


DSL Subscriber-Side Loopbacks

EOC Loopbacks

The EOC Loopback is used for troubleshooting IDSL links. Using a small portion of the ISDN band called the *Embedded Operations Channel* (EOC) is helpful in isolating IDSL line problems by allowing you to test the various segments between the CE150 and the CPE that comprise an IDSL loop. Using the EOC, each network element (that is, each segment or hop) reports its presence and is counted by the CE150. With the number of elements (number of hops) between the CE150 and the CPE known, specific network elements can then be placed into loopback mode by addressing the test command to the device that is one, two, three (or more) elements downstream from the CE150. The following diagram illustrates the concept. Note that the endpoint CPE is always referred to as Hop 0 or Element 0, regardless of how many intervening elements may be present.

The IDSL Module includes a CPE monitor that detects the presence or absence of a CPE through the EOC, and a standard IDSN network element counter. The monitor sends both standard EOC messages and a Copper Mountain specific message that the CPE use to learn the configured data rate. If for some reason the EOC path fails, the CPE may still be able to determine the data rate based on receiving Copper Mountain CMCP packets. The monitor state machine is called once a second.



IDSL Loop with Multiple Network Elements

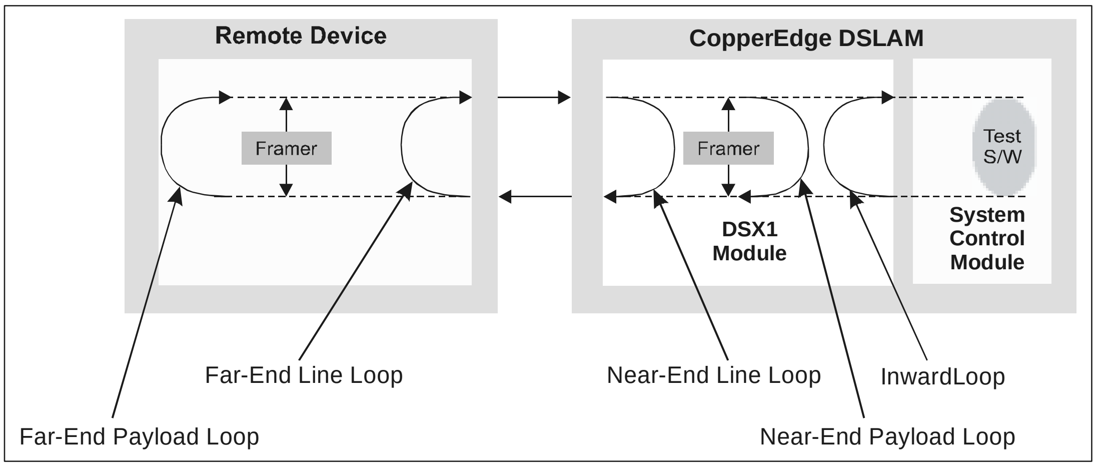
If the entire IDSL loop is intact, EOC will detect and report the presence of the CPE at the downstream end of the loop. If the link is *not* complete, EOC will attempt to count the network elements that are in place and functioning, and report the number through the cmIdslModem NetworkElements object. Thus, if you know how many hops are supposed to comprise a link, then isolating the area of the malfunction becomes much less complicated.

Similarly, if the CPE at the end of an IDSL link is able to train, but other performance factors are degraded or marginal, the SCM-DSLModuleEOC test lets you command any of the preceding network elements into loopback mode. By looping back from successive hops, you should be able to isolate the link segment where the problem is occurring. For details on configuring a CE150 for EOC loopback testing, see the cmLoop MIB description in the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual.

DS1 Loopbacks

For DS1 boards, you must configure loopbacks through the dsx1Config group. (See the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual.)

Five types of loopbacks are possible for DS1 boards: inward loops, near-end payload loops, near-end line loops, far-end line loops, and far-end payload loops. For the inward loop test, the far-end line loop test, and the far -end payload loop test, the CopperEdge must generate test packets, using a generator on its System Control Module. For the near-end line loop test and the near-end payload loop test, a remote device must generate the test packets and send them to the CopperEdge.



DS1 Loopback Modes

After setting the loopback tests for DS1 boards with the dsx1ConfigTable group of objects, you are ready to send the test packets and then read the results of the tests, using the cm-Loop group of objects.

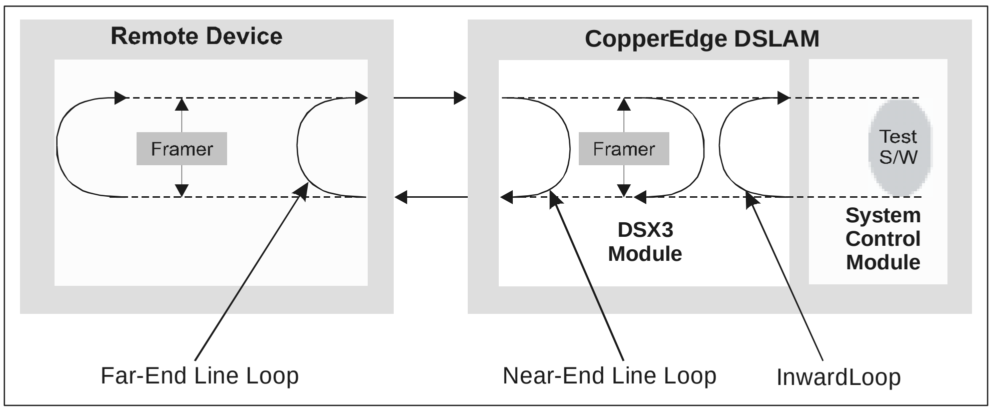
Setting up loopback and then starting, interrupting, and completing loopback tests are trapped events. As such, all of them are reported to the trap receiver.

The current CE150 product release does not support multiple simultaneous loopback tests. Only one test can be run at any one time. Moreover, the CE150 loopback test software makes no assumptions and has no prior intelligence about the loop under test. If the loop is open, even if only temporarily because up- or downstream equipment is busy, the test will run but the CE150 will report errors.

DS3 Loopbacks

For DS3 boards, either ATM or Frame, you must configure loopbacks through the dsx3Config group. (See the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual.)

Three types of loopbacks are possible for DS3 boards: inward loops, near-end line loops, and far-end line loops. For the inward loop test and the far-end line loop test, the CopperEdge must generate test packets, using a generator on its System Control Module. For the near-end line loop test, a remote device must generate test packets and send them to the CopperEdge.



DS3 Loopback Modes

After setting the loopback tests for DS3 boards with the dsx3ConfigTable group of objects, you are ready to send test packets and then read the results of the tests using the cmLoop group of objects.

Setting up loopback and then starting, interrupting, and completing loopback tests are trapped events. As such, all of them are reported to the trap receiver.

The current CE150 product release does not support multiple simultaneous loopback tests. Only one test can be run at any one time. Moreover, the CE150 loopback test software makes no assumptions and has no prior intelligence about the loop under test. If the loop is open, even if only temporarily because up- or downstream equipment is busy, the test will run but the CE150 will report errors.

Evaluating SDSL Loops

The CE150 also includes a special test capability which examines certain analog properties of the SDSL loop. The cmSDSLTest group (described in the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual) provides two tests that are useful for evaluating and troubleshooting SDSL loops. The SeekMaxRate test can help to establish the fastest data rate that a specified DSL loop will sustain with a connected CPE. The Loop Profile test examines the overall condition of the loop between physical ports on the CE150 and their respective CPE. The Loop Profile test can also be used to verify the integrity of the SDSL loop between the CE150 and the first 500 feet of the loop, typically the portion of the loop within the CO itself.

Typically, a loop test is run at the time of installation in order to establish a baseline for future assessment. Data from the test is kept on file and can then be used to perform line-quality analysis or for troubleshooting if needed.

To check continuity over the entire length of the SDSL loop, you need the cooperation of an assistant at the CPE site who can communicate in real time with the person operating the CE150. To perform this test, follow these steps:

1. Tell the assistant at the CPE site to disconnect the DSL line from the CPE, thus opening the loop.
2. At the CE150 console, enter the following command:

   ```
   CRAFT> set cmsdsltest [pii of the sdsl port]
   type=type_of_test action=start
   ```
3. At the CE150 console, wait five or ten seconds, then enter a **get cmsdsltest** command.

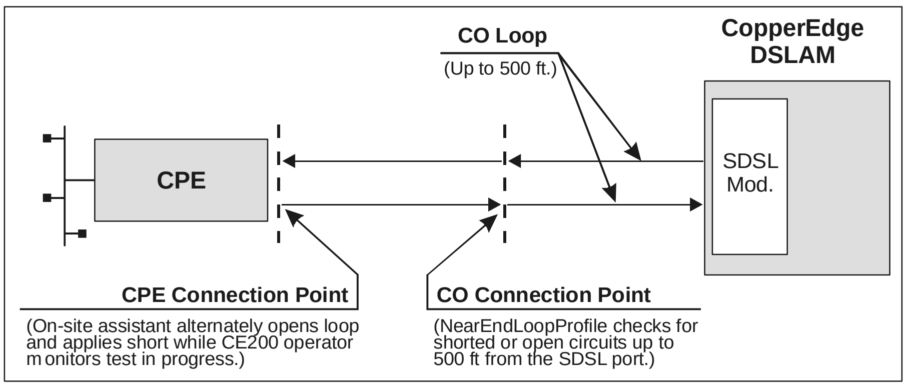
   Observe and record the value displayed in the ContinuityTest-Result field.
4. At the CE150 console, while observing the value displayed in the ContinuityTestResult field, tell the assistant at the CPE site to apply a short between pins 2 and 3 of the RJ-11 plug at the end of the DSL line.
5. At the CE150 console, wait five seconds and repeat the **get cmsdsltest** command.

   There should be a perceptible change in the value of the integer displayed in the ContinuityTestResult field; observe and record the new value.
6. At the CE150 console, while observing the value displayed in the ContinuityTestResult field, tell the assistant at the CPE site to remove the short from the RJ-11 plug at the end of the DSL line.

7. At the CE150 console, wait five seconds and repeat the **`get cmsdsltest`** command.

   The value of the integer displayed in the ContinuityTestResult field should return to the value you noted in step 3.

When using the cmSDSLTestTable for troubleshooting, the absolute numerical values displayed in the above procedure have no meaning in themselves, and will vary depending on the length of the loop and a number of other physical and electrical factors. The significant points are the change in the value as the line is monitored while being opened or shorted, and whether and by how much the resulting values deviate from the baselines recorded at the time of installation. The following diagram shows the cmSDSLTest function.



SDSL Loop Test Functionality

Built-in Performance Monitors

In addition to hard failures, the CE150 also collects information that can be helpful in diagnosing marginally performing lines or other link-related hardware. Statistics related to SDSL loops and statistics related to LAN and WAN links are collected as part of the routine operation of the system. You can configure the threshold points at which the various indicators trigger an alert. As part of a regular program of system maintenance, these statistics can help to provide early warning of link-related problems.

Statistics on system performance can also be collected for selected time intervals using the Bulk Statistics feature of the cmMaintCmd group (described in the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual).

IMUX Configuration Issues

For setting up IMUX bundles and placing ports in bundles, the CE150 offers a number of features to help identify and isolate configuration and/or wiring errors. Two types of warnings may be received:

- *Error messages*—These messages provide an immediate indication that some configuration rule has been violated. For example, if an operator, using cmBundle, tries to assign a port that has already been configured with a netmodel or a filter to a bundle.
- *TrapEvent and TrapAlarm Tables*—These tables indicate an improper step in a procedure. For example, an alarm occurs when a CPE that is not IMUX-capable trains on a bundled DSL port, such that the CE150 cannot change its status to Active (ready to receive and transmit data).

Problems Configuring Multilink Bundles

You will see an error message if you try to place ports in an IMUX bundle incorrectly and the CPE's ports are already trained. CE150 DSL ports that have already been configured as individual interfaces, and ports that have mismatched End-PointIDs, will cause the CE150 to generate error messages. The error messages related to IMUX configuration are listed below. (For the full configuration procedure for IMUX bundles, see *Configuring an IMUX Bundle* on page 100.)

Multilink Bundle Error Messages

Message	Description	Action
CPE is not IMUX capable	You cannot assign a CPE to a bundle unless it has IMUX software in its code. When a CPE trains on a port, the CE150 knows whether it is IMUX capable.	
IMUX CPE has EndPointID inconsistent with existing members	You cannot assign a port to a bundle when the CPE trained on the port has an EndPointID that does not match the EndPointID of a port already in the bundle. All ports in a bundle must have the same EndPointID.	Enter the command **`getall cmEndPointPort`** to list all End-PointIDs in the CE150, starting with the first port on the first line card. Create a new bundle, and then either switch the mismatched port or attach the correct phone to it. Finally, create other ports for the other phone lines to the bundles.

Multilink Bundle Error Messages

Message	Description	Action
IMUX CPE has EndPointID that matches member in different bundle	You cannot assign a port to a bundle when the CPE trained on the port has an EndPointID that matches the EndPointID of a port in another bundle. All ports in a bundle must have the same EndPointID.	Enter the command **`getall cmEndPointPort`** to list all of the EndPointIDs in the CE150, starting with the first port on the first line card. Either switch the port to the other bundle or attach the correct phone line to the port. Then add another port to the proper bundle and attach the correct phone line.
Member PII must be a DSL port	You cannot assign a WAN or Ethernet port to a bundle.	
Member PII must be on same shelf as BundlePII	This error message does not apply to the CE150.	
Member Port must have NetModel set to None to be added to bundle	You must set the netmodel for the IMUX bundle. You cannot set the netmodel separately for each ports in the bundle.	
Member Port must not be configured in cmDHCP table	You must set the IMUX bundle for Dynamic Host Configuration Protocol. You cannot separately configure each port in the bundle for DHCP.	
Member Port must not be configured in cmFilter table	You must set the filters for the IMUX bundle. You cannot set filters separately for each port in the bundle.	
Member Port must not be configured in cmMember table	You must assign the IMUX bundle to a user group. You cannot separately assign each port in a bundle to a group.	
Member Port must not be under loopback testing	You cannot assign a port to a bundle when it is in loopback testing. Similarly, you cannot perform loopback testing on a port already in a bundle.	To perform testing on a port in a bundle, remove it from the bundle, perform the test, and put it back in the bundle.
PII value must be a DML Bundle	The PII for an IMUX bundle represents a logical, not a physical, interface. It must have the format, 1.51.n, where n is any number from 1 to 63 inclusive.	
Port cannot be configured into multiple entries in a bundle	A bundle can have as many as four ports assigned to it. Each port must have a different PII. The ports can be on the same line card or on different line cards.	
Port is configured in a different bundle	You tried to assign the same port to different bundles. You cannot assign the same port to different bundles.	Enter the command **`getall cmBundle`** to list all IMUX bundles on the CE150 and all ports assigned to them.
Valid RowStatus values: Active, Destroy	The default value for RowStatus is Active. The only other acceptable value, Destroy, allows you to delete the bundle. You can delete the bundle without deleting ports from the bundle.	

Problems during CPE Training

If there is no CPE connected and trained on a port, any associated configuration and wiring errors will not be immediately apparent. It is only when a CPE is connected and attempts to train that the EndPointID objects are reported and inconsistencies can be detected. Without a connected and trained CPE, the CE150 simply accepts any assigned ports as part of the IMUX bundle, registers their EndPointIDs as 0.0.0.0.0.0.0.0.0.0.0, and sets the port status to WaitForAdd.

Later, if a CPE is connected and attempts to train, the CE150 can determine if the CPE is IMUX capable, and if it is, it can then compare EndPointIDs to see if they match. If the EndPointIDs of the ports in a bundle do not match, the CE150 sends an event to the TrapEvent log and an alarm to the TrapAlarm log. If this occurs, the ports remain in their previous WaitForAdd status until the conflict between the physical wiring and the logical bundles is resolved.

The TrapAlarm messages resulting from this conflict are listed below. To review the configuration of existing bundles and their member ports and isolate the source of the conflict, use the cmEndPointPort group. (See the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual.)

CPE Training Error Messages

Message	Description	Action
EndPoint Conflict, CPE in bundle is not IMUX capable	A CPE has trained on a bundle that does not have the appropriate IMUX software. Typically, the CPE has a single port instead of multiple ports.	
EndPoint Conflict, IMUX CPE has EndPointID inconsistent with existing members	When the CPE attached to the port trains, the CE150 discovers that it already has an active port in the bundle with a different EndPointID.	Either replace the port in the bundle, making sure that you have the correct phone line attached, or keep the port but switch the phone line.
EndPoint Conflict, IMUX CPE has EndPointID that matches member in different bundle	When the CPE attached to the port trains, the CE150 discovers that it has a CPE with the same EndPointID attached to a port in another bundle.	Either switch the mismatched ports between the bundles, or switch the phone lines attached to the ports.

Diagnostic Port

The Diagnostic port on the front panel of the System Control Module is for CopperMountain personnel. If you have a problem with your system, a Copper Mountain Technical Support engineer may ask you connect to the Diagnostic port to the serial port of a local computer. The Diagnostic port does provide access to certain events related to the functioning of the CE150, but not in a form readily usable by an operator.

N O T E

Copper Mountain recommends that you not connect to the Diagnostic port except as directed by Copper Mountain Customer Support. Restarting a computer with one of its serial ports connected to the Diagnostic port may cause the CE150 to reset. Also, certain service options that are available only through the Diagnostic port (such as options that facilitate recovery from software lockups) may cause undesirable results.

Syslog

In certain situations, it can be helpful to collect diagnostic information from a number of different CopperEdge units (CE200 or CE150), such as when multiple units are exhibiting the same or similar kinds of symptoms. To facilitate collecting this data at a central location, the CopperEdge unit can be configured as a UNIX Syslog client.

With the Syslog capability, diagnostic information from a number of CopperEdge units, which was previously available only at each unit's control module Diagnostic port, can be sent directly to a file on a UNIX-based server (Solaris or Linux) in standard Syslog message format.

While the Syslog capability requires that you configure the CopperEdge units and UNIX server as clients, the log file should not be considered a user-accessible function.

Since Syslog messages are identical in content and format to those at the CopperEdge Diagnostic ports, the same cautions apply: Syslog message information cannot be reliably interpreted by an operator, and should only be retrieved with the active participation of Copper Mountain Technical Support. Data from the log can then be returned to the factory where the contents can be analyzed by Copper Mountain technical staff.

N O T E

Syslog messages occur as the result of certain normal operational events. The mere presence of data in the Syslog should not be construed as indicative of trouble with the system, and should not trigger a service call.

Configuring the Syslog Server

To implement Syslog capability on your system, you must first identify and configure a suitable UNIX machine as your Syslog file server. If you need more information on how to do this, consult the on-line UNIX manual:

- *man syslog*—Provides an overall description of the function and lists the available options and modifiers
- *man syslogd*—Describes the syslog daemon and lists its available options
- *man syslog.conf* —Provides server configuration details

When your Syslog server is operating and accessible to the network, client machines can be configured as described in the subsequent section.

Once you have begun collecting information in the Syslog, managing the records becomes a significant consideration. By default, most UNIX machines will purge or truncate log files at preset intervals. It is important that your server's "flushing" interval be set at a large enough value to allow a meaningful body of data to be gathered. We recommend at least one week.

Syslog Client Configuration

Four objects in the cmFile MIB group (SysLogAddr, SysLogPort, SysLogFacility, and SysLogPriority) control the Syslog capability on the CopperEdge unit. When configuring the CopperEdge unit as a Syslog client, all you need to do is enter the IP address of the server. In most cases, the default settings of all of the other objects should be the appropriate ones. For example:

```
CRAFT> set cmfile syslogaddr=209.141.14.4
```

For more information about other Syslog configuration options for the CE150, refer to the cmFile MIB definition in the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual.

CPE Message Log Table

If you need to perform diagnostics for an IAD CPE (such as a CR408, CR508, or CR508T), you can log debug messages to a CPE log table, which you can view from the Craft prompt.

To log messages, you will use two MIB groups: cmCpeIADTable and cmCpeLogTable.

- cmCpeIADTable—Create and delete the log table entries using the LogAction, LogType, LogLevel, and LogDuration objects.
- cmCpeLogTable—View up to 1000 logged message entries. The first entry contains the oldest information.

In the following examples, we will turn on message logging and generate some messages.

To start logging messages:

1. First check the cmCpeBoardTable to see if the CPE supports cmCpeLogTable.

```
CRAFT>  get cmcpeboard [cpe:1.5.2] groupmap
Group: cmCPEBoardTable
Instance: [CPE:1.5.2]
GroupMap            =  3a.40.39.91.92.93.94.9b.
                       a5.a0.a7
```

   In the GroupMap, a5 indicates cmCpeLogTable.

2. Then check the cmCpeIADTable to see how the LogAction object is set.

```
       CRAFT> geta cmcpeiad
Index           CurrentIpAddr     CurrentCAIpAddr    SavedIpAddr
SavedCAIpAddr    NumVoicePorts     NumConnections     Command
TOSByte          LogAction         LogType            LogLevel
LogDuration     TotalMsgsLogged    Trace1Mask        Trace2Mask


Instance: [CPE:1.13.2]
CPE:1.13.2       10.10.1.5         10.10.1.1          10.10.1.5
10.10.1.1        8                 16                 None
10               None              0                  0
30               0                 0                  0
```

3. Since LogAction=None, there should be no messages logged in the cmCpeLogTable.

```
CRAFT> geta cmcpelog
No more instances
```

4. Turn on the debug message logging feature for the duration of the timer, set in the LogDuration object:

```
CRAFT> set cmcpeiad [cpe:1.5.2] logaction=startdebug
Set Successful
```

   When the timer expires, the LogAction object is automatically set to StopDebug and messages are no longer logged.

5. Select a log type by entering either the text or the bit number for the text.

```
CRAFT> set cmcpeiad [cpe:1.5.2] logtype=dhcp
OR
CRAFT> set cmcpeiad [cpe:1.5.2] logtype=4
Set Successful
```

6. Specify the message levels to be logged by entering either the text or the bit numbers for the text.

```
CRAFT> set cmcpeiad [cpe:1.5.2] loglevel=0+1+2+
OR
CRAFT> set cmcpeiad [cpe:1.5.2]
loglevel=info+error+detail+other
Set Successful
```

7. (Optional) You can change the timer for logging messages (for example, from 30 minutes to 10 minutes):

```
CRAFT> set cmcpeiad [cpe:1.5.2] logduration=10
Set Successful
```

8. For this example, we can generate some messages to be logged by changing the link's administrative state to Down and then back to Up. This forces a resend of the DHCP request, causing messages to be logged.

```
CRAFT> set iftable [1.5.2] admin=down
Set Successful
CRAFT> set iftable [1.5.2] admin=up
Set Successful
```

9. Check the IAD table and note that the TotalMsgsLogged object has changed from 0 to 5.

```
        CRAFT> geta cmcpeiad
Index             CurrentIpAddr       CurrentCAIpAddr      SavedIpAddr
SavedCAIpAddr      NumVoicePorts       NumConnections        Command
TOSByte            LogAction           LogType               LogLevel
LogDuration       TotalMsgsLogged     Trace1Mask           Trace2Mask


Instance: [CPE:1.13.2]
CPE:1.13.2         10.10.1.5           10.10.1.1            10.10.1.5
10.10.1.1          8                   16                   None
10                 StartDebug          4                    0+1+2+3
30                 5                   0                    0
```

To see more details:

```
CRAFT> get cmcpeiad [cpe:1.5.2]
Group: cmCpeIADTable
Instance: [CPE:1.5.2]
Index              =  CPE:1.5.2
CurrentIpAddr      =  10.10.1.5
CurrentCAIpAddr    =  10.10.1.1
SavedIpAddr        =  10.10.1.5
SavedCAIpAddr      =  10.10.1.1
NumVoicePorts      =  8
NumConnections     =  16
Command            =  None
TOSByte            =  10
LogAction          =  StartDebug
LogType            =  DHCP(4)
LogLevel           =  INFO(0)+ERROR(1)+DETAIL(2)
                      +OTHER(3)
LogDuration        =  30
TotalMsgsLogged    =  5
Trace1Mask         =  0
Trace2Mask         =  0
```

10. Display the message entries in the log table:

```
        CRAFT> geta cmcpelog
Index              MsgNbr             MsgType            TimeStamp
Message

Instance: [CPE:1.13.2, 1]
CPE:1.13.2         1                  DHCP               587565
Sending DHCP reques

Instance: [CPE:1.13.2, 2]
CPE:1.13.2         2                  DHCP               587575
Got a DHCP Reply

Instance: [CPE:1.13.2, 3]
CPE:1.13.2         3                  DHCP               587575
ipaddr 10.10.1.5

Instance: [CPE:1.13.2, 4]
CPE:1.13.2         4                  DHCP               587575
iad_hdia_enable is

Instance: [CPE:1.13.2, 5]
CPE:1.13.2         5                  DHCP               587575
caipaddr 10.10.1.1

No more instances
```

You can see that five new message entries have been added to the log table. The Geta command does not always show the entire message, so perform a Get to see more details for some of the entries.

```
CRAFT> get cmcpelog [cpe:1.5.2,1]
Group: cmCpeLogTable
Instance: [CPE:1.5.2, 1]
Index                =  CPE:1.5.2
MsgNbr               =  1
MsgType              =  DHCP
TimeStamp            =  587565
Message              =  Sending DHCP request

CRAFT> get cmcpelog [cpe:1.5.2,4]
Group: cmCpeLogTable
Instance: [CPE:1.5.2, 4]
Index                =  CPE:1.5.2
MsgNbr               =  4
MsgType              =  DHCP
TimeStamp            =  587575
Message              =  iad_hdia_enable is 1
```

To stop logging messages:

Turn off the message logging feature:

```
CRAFT> set cmcpeiad [cpe:1.5.2] logaction=stopdebug
Set Successful
```

The log table still displays the messages that were already logged. However, any new messages will not be added to the log table.

To delete entries from the message log:

Set the LogAction object to delete the entries.

```
CRAFT> set cmcpeiad [cpe:1.13.2] logaction=deletedebug
Set Successful
```

The TotalMsgsLogged object has been reset to 0.

```
CRAFT> geta cmcpeiad
Index              CurrentIpAddr       CurrentCAIpAddr     SavedIpAddr
SavedCAIpAddr       NumVoicePorts       NumConnections      Command
TOSByte             LogAction           LogType             LogLevel
LogDuration        TotalMsgsLogged     Trace1Mask          Trace2Mask


Instance: [CPE:1.13.2]
CPE:1.13.2          10.10.1.5           10.10.1.1           10.10.1.5
10.10.1.1           8                   16                  None
10                  DeleteDebug         4                   0+1+2+3
30                  0                   0                   0
```

All existing messages have been deleted from the cmCpeLog-Table.

```
CRAFT> geta cmcpelog
No more instances
```

Set the LogAction object to None:

```
CRAFT> set cmcpeiad [cpe:1.5.2] logaction=none
Set Successful
```

Restart Options

As mentioned elsewhere, the CE150 is designed to provide uninterrupted service, 24 hours a day. Resetting or restarting the system or any of its modules is service-affecting, and should only be done when warranted by a failure that cannot be resolved any other way.

Nevertheless, situations may arise that can only be resolved by resetting a subscriber CPE, a module in the CE150, or by rebooting the CE150 itself.

CPE Soft Restart

If you suspect that a particular subscriber CPE is experiencing a software fault, it may be possible to recover by restarting the unit from the CE150. When this command is issued, the CPE should restart and retrain on the DSL port, and reload its software. Normal data transmission to and from the CPE will be interrupted while the restart sequence completes.

To restart a CPE, enter the following command:

```
CRAFT> set cmcpeboard [cpe:pii] command=restart
```

In this case, the PII is the permanent interface identifier of the CE150 DSL port to which the specified CPE is connected.

Module Soft Restart

You can also perform a soft restart on the CE150 System Control Module, any of the DSL modules, or a DS3 Frame module (but *not* to Buffer Control modules, to V.35 WAN modules, or DS3 ATM modules). If you suspect a software fault within a specific module, it may be possible to recover by restarting *only* the specific module. In some cases this may be a less drastic solution than restarting the entire chassis. That is, it might affect fewer subscribers (as in restarting only a single DSL module). Also, restarting a single module generally takes less time than a complete reboot.

To restart a specific module, enter the following command:

```
CRAFT> set cmboard [pii] command=restart
```

In this case, the PII consists only of the shelf and slot number. For example, a DSL module in slot 5 would be specified as [1.5].

N O T E

A Restart command directed to a module that does not support individual module restart may return a "`Set Successful`" message, but in fact the command will be ignored.

Line Card Restart (Hardware Restart)

If, for some reason, you are unable to restart a DSL module using the normal software restart option, (**`set cmboard`**) a hardware restart is available through the CopperCraft interface that will achieve the same result.

To perform a hardware restart of a DSL module, connect to the system control module (in system equipped for redundancy, this must be the current Primary control module) through the CopperCraft interface and issue the following command:

```
CRAFT> lcrestart n.n
```

where *n.n* indicates the shelf and slot number of the target DSL module. Note that this command cannot be abbreviated.

SCM-3 Reset Switch

If the software restarts do not work and your CE150 is equipped with a System Control Module-3 (SCM-3), press the Reset switch on the front panel of the module. It resets the System Control Module, Buffer Control Module, and any WAN modules in a complex.

System Configuration

If you suspect the CE150's configuration data has become unstable or corrupt, you may need to reload the backup configuration file from a remote file server using cmMaintCmd and its ConfigRestore option as described in *Restoring a Backed Up Configuration* on page 111. Implementing the recovered data file, however, does require a full restart of the CE150 as discussed in the next section.

The cmSystem ReadConfig command is not supported in this release.

System Soft Restart

In a major software fault occurs, there may be no alternative to restarting the system. To restart/reboot the CE150, enter the following command:

```
CRAFT> set cmsystem command=restart
```

Using SCMRestart

In certain relatively rare cases, a system failure could be accompanied (or caused) by an "out of control buffers" state. If you have tried unsuccessfully to soft restart the Control Module as described above (**`set cmboard [1.2] command=restart`**) and the system (**`set cmsystem command=restart`**), there is one other alternative recovery procedure: SCMRestart.

The SCMRestart command does not require any control buffers to be available. To run this command, enter:

```
CRAFT> scmrestart
```

Note that the *entire word* must be entered, but it isn't case-sensitive. You must have Provisioning or greater privileges to execute this command.

Removing or Replacing Modules

This section describes the procedures for installing or replacing the System Control Module, Buffer Module, WAN modules, DSL modules, and Power Supply modules.

C A U T I O N

*All CE150 modules contain static-sensitive devices. If you must remove or handle modules for **any** reason, observe standard ESD precautions (such as using ground straps for personnel and equipment, and storing removed modules in anti-static bags). If you are unsure of the necessary precautions, contact Copper Mountain Technical Support for assistance.*

System Control Module

Replacement of a System Control /WAN Module is service-affecting, with consequent interruption of service for all subscribers served by this CE150. Except in cases of emergency (system down), we recommend that you replace the module during a scheduled maintenance period at off-peak hours.

To replace a SCM/WAN module:

1. If the installed System Control Module is operational:
   - Set FTP to Binary mode and copy the Configuration file to a workstation.
   - Or, back up the existing configuration file as described in *Backing Up the Saved Config File* on page 110.
2. Power down the system and disconnect all front panel cables from the System Control/WAN Module.
3. Use a Phillips screwdriver to loosen the two screws securing the System Control/WAN Module.
4. Lift the module extraction levers to gently separate the System Control Module from its connector inside the CE150.

   When the module is free, remove it from the CE150 and transfer it to a suitable anti-static container.
5. Remove the new System Control Module from its anti-static bag, and install in the empty CE150 System Control slot.
6. Gently but firmly seat the module into its connector inside the CE150.

   If the module does not readily mesh with the connector pins, do not force it; pull the module out and try again.

7. When the module is seated in the connector, ensure that the extraction levers lay flat (module-inserted position), and use your fingers to screw in the two retainer screws.

   Finger tighten only; do not over-torque.

8. Reconnect the front-panel cables disconnected in step 2.

9. Apply power to the system and wait for boot-up to complete.

10. Connect to the Craft port and use a terminal emulation program to log in to the CE150.

11. Configure the CE150 to talk to the rest of the network:

    a) Set the netmodel, IP address, and netmask of the CE150.

    ```
    CRAFT> set cmiface [1.3.1] netmodel=ip
    ipaddress=xxx.xxx.xxx.xxx netmask=255.255.255.xxx
    ```

    b) Set the default route for this CE150.

    ```
    CRAFT> set iproute [0.0.0.0] mask=255.255.255.xxx
    nexthop=xxx.xxx.xxx.xxx
    ```

    c) Set the SNMP command strings.

    ```
    CRAFT> set cmoperator [public] context=snmp
    privilege=view

    CRAFT> set cmoperator [private] context=snmp
    privilege=provision
    ```

    d) Save the new configuration.

    ```
    CRAFT> set cmsystem command=save
    ```

12. If necessary, load the software.

13. Restore the saved configuration:

    - Set FTP to Binary mode and copy the Configuration file from the workstation.
    - Or, refer to *Restoring a Backed Up Configuration* on page 111 to reinstate the saved system configuration.

14. Restart the system.

    ```
    CRAFT> set cmsystem command=restart
    ```

N O T E

If you are upgrading to a Buffer Control Module 2, you must also upgrade to a System Control Module 3. The BCM-2 will not work with a SCM-1 or SCM-2.

Buffer Control Modules

Replacement of the Buffer Control module is service-affecting, with consequent interruption of service for all subscribers served by this CE150. Except in cases of emergency (system down), we recommend that you replace any modules during a scheduled maintenance period at off-peak hours.

N O T E

If you are upgrading to a Buffer Control Module 2, you must also upgrade to a System Control Module 3. The BCM-2 will not work with a SCM-1 or SCM-2.

To replace a Buffer Control Module:

1. Power down the system and disconnect any front panel cables from the affected module.
2. Use a Phillips screwdriver to loosen the two screws securing the module to be replaced.
3. Lift the module extraction levers to gently separate the module from its connector inside the CE150.

   When the module is free, remove it from the CE150 and transfer it to a suitable anti-static container.
4. Remove the new module from its anti-static bag and insert it into the empty slot in the CE150.
5. Gently but firmly seat the module into its connector inside the CE150.

   If the module does not readily mate to the connector, do not force it; back the module out and try again.
6. When the module is seated in the connector, ensure that the extraction levers lay flat (module-inserted position), and use your fingers to screw in the two retainer screws.

   Finger tighten only; do not over-torque.
7. Reconnect the front-panel cables disconnected in step 1.
8. Apply power to the system and wait for boot-up to complete.

DSL Line Modules

Replacement of any DSL line module--SDSL, ADSL (G.dmt or G.lite), or T1--is service-affecting to the ports served by that module. Except in cases of emergency (system down), we recommend that you replace any modules during a scheduled maintenance period at off-peak hours. Unlike the other modules, however, you can reliably replace DSL line modules without powering down the CE150.

N O T E

*Before you remove a module that will **not** be replaced by another module of the same type, change the netmodel to None for all VCs, ports, and IMUX bundles associated with that module.*

To replace a DSL line module:

1. Use a Phillips screwdriver to loosen the two screws securing the module to be replaced.
2. Lift the module extraction levers to gently separate the module from its connector inside the CE150.

   When the module is free, remove it from the CE150 and transfer it to a suitable anti-static container.
3. Remove the new module from its anti-static bag, and insert it into the empty slot in the CE150.
4. Gently but firmly seat the module into its connector inside the CE150.

   If the module does not readily mate to the connector, do not force it; back the module out and try again.
5. When the module is seated in the connector, ensure that the extraction levers are lying flat (module-inserted position), and use your fingers to screw in the two retainer screws.

   Finger tighten only; do not over-torque.
6. After the new DSL module has been installed, you must reconfigure the data rate of each DSL port as appropriate; connected CPEs will follow the rate of the DSL interface. For example:

   ```
   CRAFT> set hdslmodem [1.4.1] datarate=416
   ```

AC or DC Power Modules

Replacing an AC or DC Power module is service-affecting, with consequent interruption of service for all subscribers served by this CE150. Except in cases of emergency (system down), we recommend that you replace the module during a scheduled maintenance period at off-peak hours.

To replace the power module:

1. Power down the CE150.
2. Remove and replace the module using the general technique described in *Buffer Control Modules* on page 213.
3. Power up the CE150.

Copper Mountain Service and Support

The Copper Mountain Customer Service Center (CSC) is your source for assistance in diagnostics and/or problem resolution, and for information on Copper Mountain products and service.

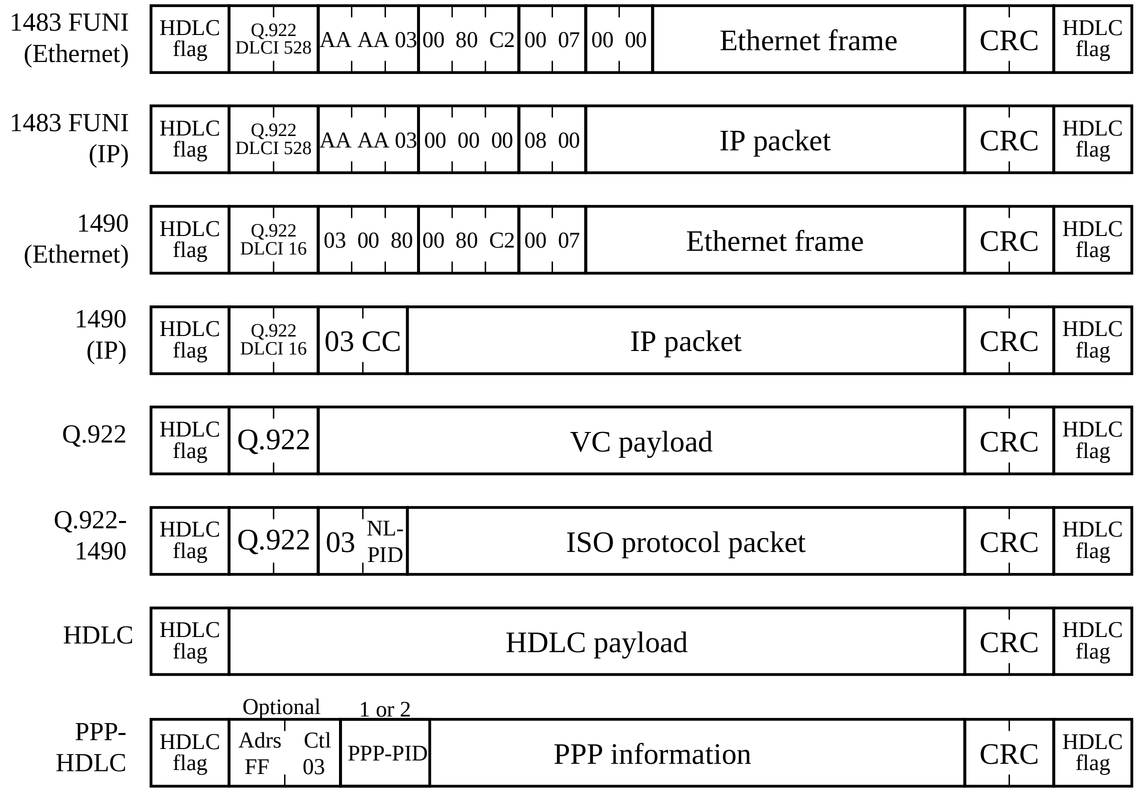
Telephone support (toll-free from any US location) is available Monday through Friday from 6 a.m. to 6 p.m., Pacific time, at 888-611-4CMN (888-611-4266).

If you suspect a failure in your CE150 or one of its modules, contact the Copper Mountain CSC for help and instructions. Should it be necessary to return any of your Copper Mountain equipment for service or repair, you will be provided with shipping instructions and the Repair Authorization numbers. These numbers should always be clearly marked on the outside of the package to ensure the fastest possible repair or replacement.

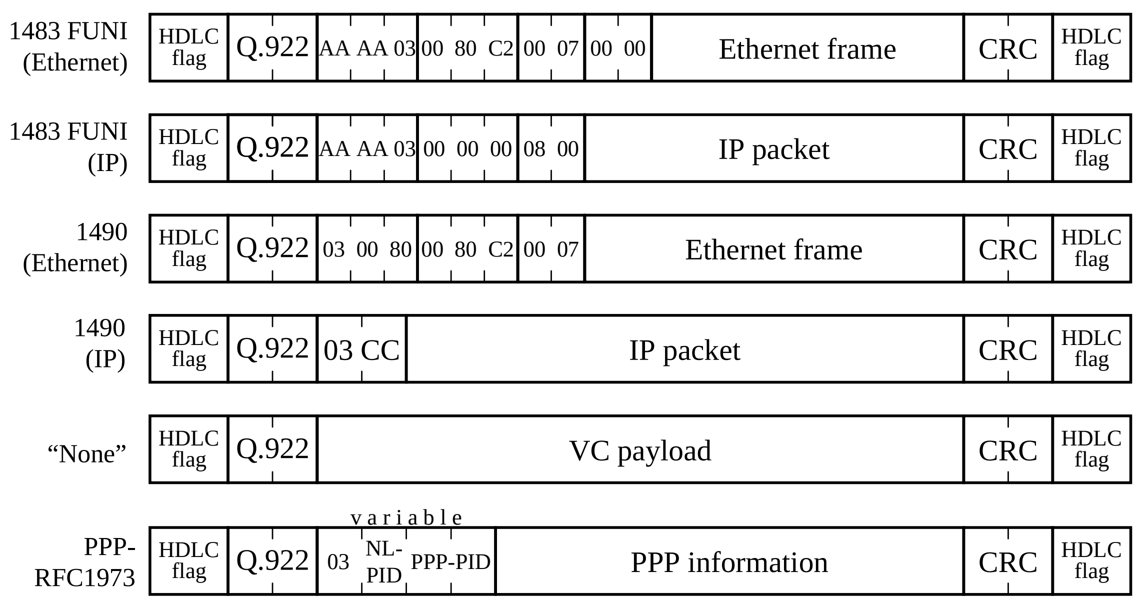
Appendix A
Overview of Data Encapsulation

As the capabilities of the CopperEdge product family have grown to embrace an ever wider pool of users and service providers, the number of supported protocols and encapsulations has grown as well. The diagrams presented below and on the following page provide a graphic representation of each of the encapsulation types supported in Version 2.1, which interfaces and networking models they apply to, and how they resemble and differ from each other.

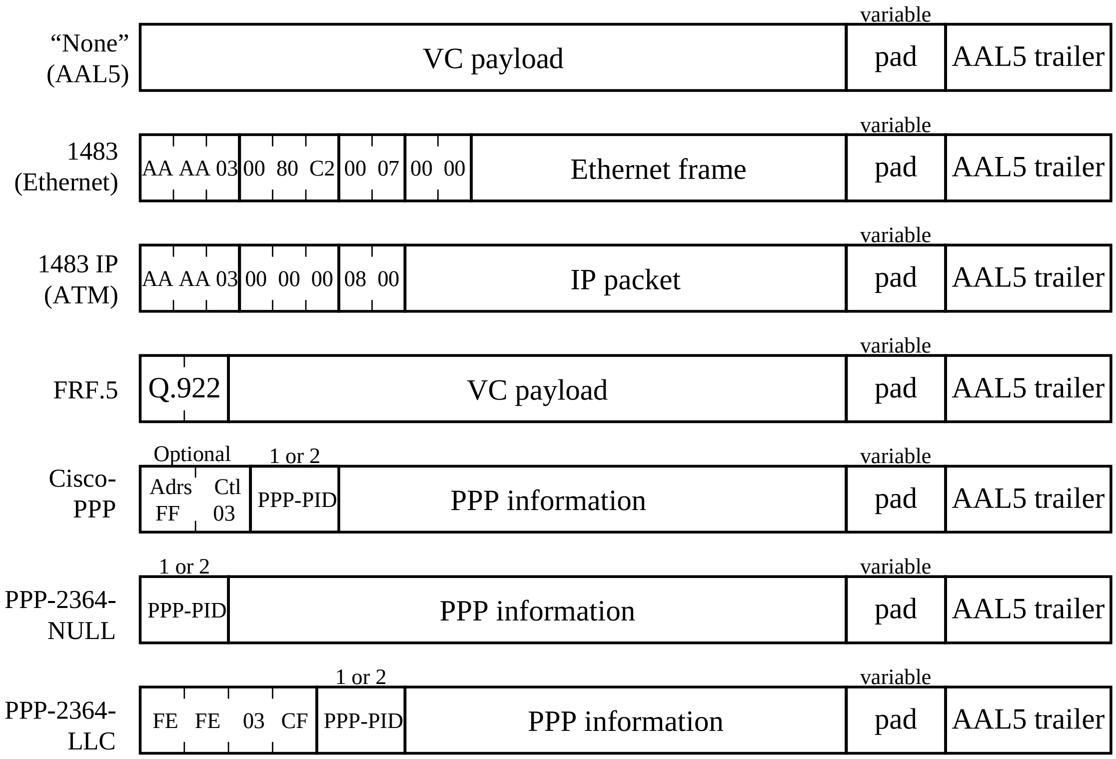
DSL Port Encapsulations



Frame Relay WAN VC Encapsulations



ATM WAN VC Encapsulations



Netmodels, Encapsulations, and Translations

It is important to distinguish between *port encapsulations* and *VC encapsulations*. The following sections describe DSL port encapsulations, DSL VC encapsulations, WAN port encapsulations, and WAN VC encapsulations. Known inconsistencies appear at the end.

DSL Port Encapsulations

For DSL ports, the port encapsulation defines whether the DSL port is a single logical interface or it supports multiple VCs. Encapsulations of HDLC or PPP-HDLC are a single logical interface and cannot support VCs. All other encapsulations support multiple VCs on a DSL port.

DSL physical ports support the following encapsulations. Note that in the cross-connect Netmodel w/ Q.922, ATM or RFC1973, DSL VCs defined with cmSubIfaceTable entries assume an encapsulation implied by the port encapsulation.

DSL Physical Port Encapsulations

DSL Port Encapsulation	Implied DSL VC Encapsulation	Valid Netmodels	Description	Notes
None	cmSubIface not allowed	None		The interface is not yet configured for data transport
RFC1483	cmSubIface not allowed	VWAN, IP, CVPN, HDIA	Funi VPI/VCI of 1/32 + 1483(MAC) = Frame Relay DLCI 528 + 1483 (MAC)	
RFC1490	cmSubIface not allowed	VWAN, IP, CVPN, HDIA	Frame Relay DLCI 16 + 1490(MAC)	ICP is optional
Q922	None	Cross-Connect	Q.922 VCs of unspecified content	Used to specify Layer-2 Frame Relay interworking modes.
ATM	None	Cross-Connect	G.lite ATM VC unspecified content	Used to specify Layer-2 ATM interworking modes.
Q922-1490	RFC1490	Cross-Connect	Q.922 VCs, each translated VC containing RFC1490 encapsulation.	Used for FRF.8 translation. Non-translated VCs can have arbitrary (unknown) encapsulation
HDLC	cmSubIface not allowed	Cross-Connect	Raw HDLC frame of unspecified content. No VCs.	Used for Cisco-PPP. Cannot allow VCs or ICP, because there is no Q.922 header.

DSL Physical Port Encapsulations

DSL Port Encapsulation	Implied DSL VC Encapsulation	Valid Netmodels	Description	Notes
PPP-HDLC	cmSubIface not allowed	Cross-Connect	PPP (RFC1662) + PPP Payload. No VCs.	Payload excludes HDLC address and control fields. No VCs or ICP, because there is no Q.922 header.
RFC1973	RFC1973	Cross-Connect	Either PPP-transparent or PPP-translation Fwd-Mode, depending on WAN encapsulation	FRF.8 translation performed when WAN encapsulation is RFC2364-null or RFC2364-LLC

Dual Netmodel

Except for HDLC and PPP-HDLC, all DSL port encapsulations (including None) allow a dual-netmodel cmIface entry on VCID=22. This means that such encapsulations imply a port-level encapsulation that supports VCs, which is ATM or RFC2364-null/ RFC2364-LLC for ATM-based DSLs, and Q.922 or RFC1973 for frame-based DSLs.

DSL VC Encapsulations

In one special case (DSL VC VCID=22), you specify a DSL VC encapsulation in a cmIfaceTable row. Such DSL VCs support these encapsulations:

DSL VC22 Encapsulations

DSL VC Encapsulation	Valid Netmodels	Description	Notes
None	Cross-Connect	Q.922 (SDSL) or AAL5 (G.lite and G.dmt)	Used for transparent forwarding
RFC1490	Cross-Connect	1490 IP or MAC frames	Used for FRF.8 translation
IP-1490	HDIA	Routed IP frames	

WAN Port Encapsulations

Currently, WAN ports do not have cmIface entries or explicit configuration. Since the hardware for each WAN port supports only one encapsulation, the encapsulation is implied by the hardware.

Frame-based WAN interfaces (such as DS3-Frame and Quad-T1) imply a WAN port encapsulation of Q.922 (Frame Relay). ATM WAN interfaces (such as DS3-ATM) imply a WAN port encapsulation of ATM. In the future, frame-based WANs may support non-Q.922 encapsulations and you will then have to explicitly configure the WAN port encapsulations.

WAN VC Encapsulations

WAN VCs (Frame Relay and ATM) support the following encapsulations. For 1483 or 1490 encapsulation, the netmodel determines whether the frame-type is IP PDU or MAC PDU.

WAN VC Encapsulations

WAN VC Encapsulation	Valid Netmodels	Frame-based VC Format	ATM VC Format
None	None, Cross-Connect	Q.922 address + payload	AAL5 payload
RFC1483	VWAN, IP, CVPN, HDIA, Cross-Connect	Q.922 address + 1483 (MAC or IP)	AAL5 + 1483
RFC1490	VWAN, IP, CVPN, HDIA	Q.922 address + 1490 (MAC or IP)	AAL5 + 1490
RFC1973	Cross-Connect	Q.922 address + RFC1973	
RFC2364-null	Cross-Connect	none	PPP payload + AAL5 trailer
RFC2364-LLC	Cross-Connect	none	LLC header + PPP payload + AAL5 trailer
FRF.5	Cross-Connect		AAL5 + FRF.5

Ethernet Port Encapsulations

Since industry standards unambiguously define a single encapsulation for each protocol, the CE150 only accepts Encapsulation=None.

Translations

For the Cross-Connect netmodel, the CE150 does not terminate protocols, but may convert headers. For non-VC DSL port encapsulations, there is only one logical interface per DSL port. For DSL port encapsulations of Q.922, RFC1973, or ATM, each DSL VC may be a separate logical interface.

DSL Port Translations

This table covers the DSL port encapsulations that do not support VCs. It describes Cross-Connects from a DSL port to a WAN VC.

DSL Cross-Connect Encapsulations

Netmodel	DSL Port Encapsulations	WAN VC Encapsulations	FwdMode	Notes
Cross-Connect	HDLC	None	HDLC-VC-payload	Used for PPP-HDLC to Cisco-PPP-ATM
	PPP-HDLC			
Cross-Connect	PPP-HDLC	RFC1973	PPP-HDLC-1973	PPP-HDLC to RFC1973

DSL VC Translations

This table covers the DSL port encapsulations that support VCs. It describes Cross-Connects from a DSL VC to a WAN VC. Since DSL VCs defined in the cmsubIface table have no FwdMode object, operators must look to the Cross-Connected WAN VC for the FwdMode, which the CE150 uses for both the DSL VC and the WAN VC. In these cases, the DSL port has a FwdMode=Per-VC, since each DSL VC has its own FwdMode.

DSL Port VC Encapsulations

Netmodel	DSL VC Encapsulations[a]	WAN VC Encapsulations	FwdMode	Notes
Cross-Connect	None	None	VC-VC-payload	payload transparent, including PPP-RFC2364.This is not really a translation.
		FRF.5	FRF.5	FRF.5
Cross-Connect	RFC1490	RFC1483	FRF.8-1490-1483	FRF.8 translate
		None	VC-VC-payload	payload transparent, including PPP-RFC2364
		FRF.5	FRF.5	FRF.5
Cross-Connect	RFC1973	RFC 1973	PPP-Transparent	Like VC-VC-payload, but with encapsulations specified and validated
		RFC 2364-null	PPP-Translation	FRF.8 translate
		RFC 2364-LLC	PPP-Translation	FRF.8 translate
		None	VC-VC-payload	payload transparent

a. [1] For DSL VCs configured with a cmSubIfaceTable entry, the VC encapsulation is implied by the DSL port encapsulation. For DSL VCs configured with a cmIfaceTable entry, the VC encapsulation is specified directly by that cmIfaceTable entry.

Appendix B
cmIface Configuration

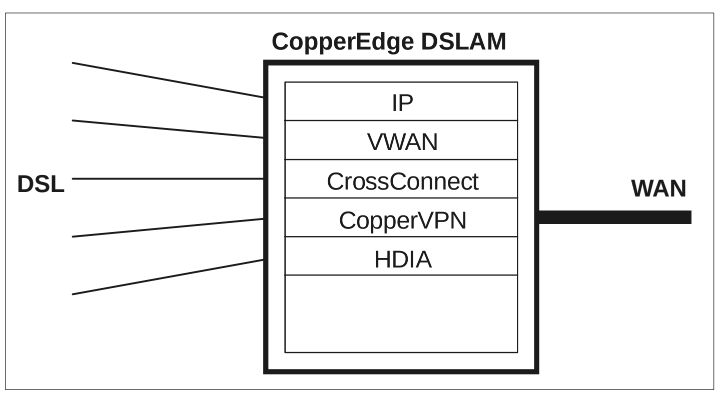
Data Forwarding Configuration

The CE150 is capable of a wide variety of protocol processing functions for user data. It requires a significant amount of operator configuration to implement functions.

Each interface on the CE150 operates in one of five major networking models (or *netmodels*):

- IP
- VWAN
- Cross-Connect
- CopperVPN
- HDIA

You can configure interfaces independently, and all netmodels may be simultaneously operating in a single CE150.

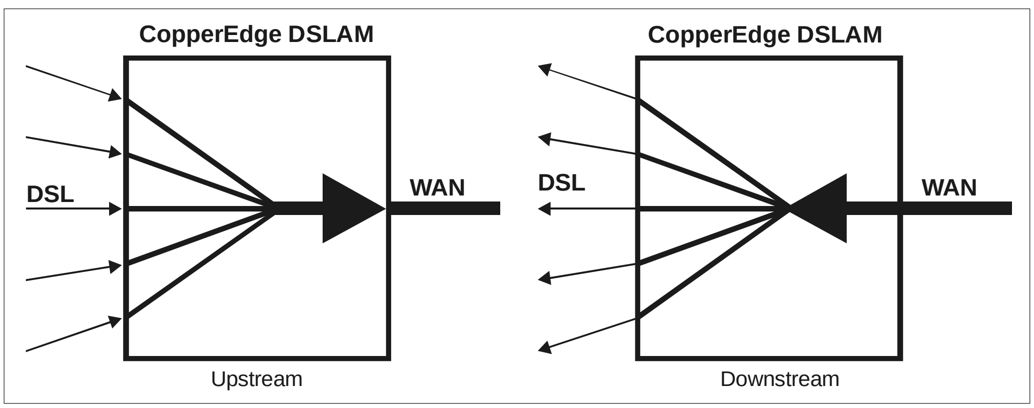


CE150 with Five Simultaneous Netmodels

In general, the netmodels behave quite differently and require very different configuration. Later in this appendix we will discuss each netmodel separately. But first we describe some common principles used in the CE150.

The CE150 is a DSLAM, or Digital Subscriber Line Access Multiplexor. Therefore, it is primarily a data forwarding and multiplexing device that connects DSL subscribers to Network Service Providers (NSPs). An NSP may be a third-party service provider, such as an Internet Service Provider (ISP), or it may be a private company providing data services to its employees or branch offices. The CE150 forwards the vast majority of user data between subscribers and their NSPs. As part of the forwarding process, the CE150 often modifies the data in some way, but very little user data actually terminates in the CE150.

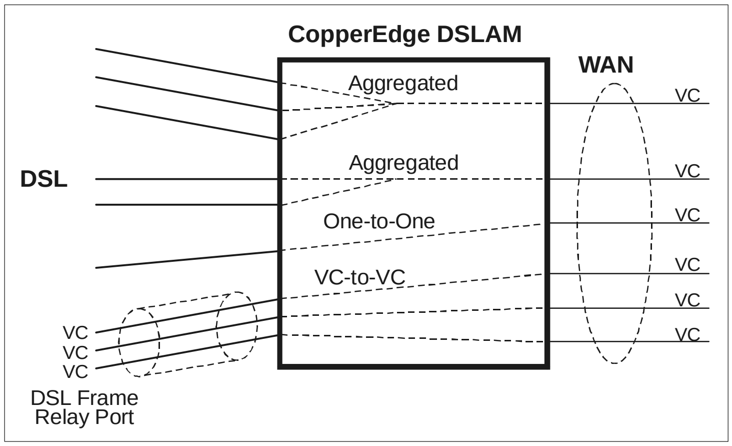
Although the CE150 can forward data between any two interfaces, most applications involve the CE150 aggregating upstream traffic from many DSL subscribers into one (or a few) multiplexed WAN interfaces, and demultiplexing downstream traffic from the multiplexed WAN interface to many DSL subscribers. A multiplexed WAN interface is typically a high-speed Frame Relay or ATM interface, with many independent virtual circuits (VCs) on it. Each virtual circuit is a logical point-to-point link to an NSP. Note that in some configurations, the WAN interface can be a CE150 Ethernet port, or even another DSL port.



CE150 with Upstream and Downstream Traffic

DSL subscribers are usually connected, at subscription time, to a single NSP, and therefore, to a single VC on the WAN interface. Thus, you can often think of CE150 configuration as connecting CE150 DSL ports to CE150 WAN VCs.

However, the CE150 does much more than just connect ports; it also performs many other important networking functions, as described in the netmodel sections. In some netmodels, many DSL ports can be aggregated together onto a single WAN VC. In other netmodels, it is only meaningful to map a single DSL interface to a single WAN VC. In these cases, the CE150 enforces this one-to-one connection requirement at configuration time. A third possibility is that the DSL port is a Frame Relay or ATM interface containing VCs within it. Then the VCs on the DSL port are connected to VCs on the WAN port.



CE150 with Aggregated and One-to-One Connections

Because many DSL ports may be connected to a single aggregated interface, you always specify connections at the DSL interface. That is, you define connections by configuring the DSL port to connect to a WAN VC. The connection is two-way; data traffic can flow in both directions. But, you always configure a connection by connecting a DSL port to a WAN VC, not by connecting a WAN VC to a DSL port.

The cmIfaceTable

At the center of all user data processing configuration is a single table: the cmIfaceTable (often abbreviated cmIface). All user-data configuration starts with the cmIface table. In some cases, several other support tables augment the cmIface table. But each support table is needed only in some configurations, and only for some functions. Different applications and interfaces use different support tables.

The cmIface table consists of many rows, or instances, of cmIface entries. Each entry configures a single interface, and in some cases, connects one interface to another. Each entry consists of several objects. The exact usage of the objects is highly dependent on what kind of interface is being configured, and on which netmodel you choose for user-data processing.

In all cases, a single cmIface entry configures a DSL port, a DSL voice circuit, an Ethernet port, or a WAN virtual circuit.

Permanent Interface Identifier

The cmIface table is indexed by a Permanent Interface Identifier (PII). This means each cmIface entry is uniquely identified by its PII. A single PII can have no more than one cmIface entry. Note that PIIs are used for other purposes besides the cmIface table, so not all PIIs in the system have a cmIface entry.

PIIs are called permanent because they do not change across power cycles of the CE150, and the PII for an interface does not change as a result of any configuration of other interfaces. Each DSL port, Ethernet port, and WAN VC is uniquely identified by a PII. In some cases, the CE150 may require some enabling configuration to make the PIIs for some VCs valid; not all PIIs are necessarily valid with all configurations. The CE150 enforces any requisite configuration by disallowing the use of PIIs that are not currently valid. PIIs are unique across all interfaces in a single CopperEdge chassis.

The PII format is completely described elsewhere, but briefly, a PII is a sequence of one to four decimal numbers separated by periods. The four numbers are the chassis, slot, port, and circuit ID of the interface. For example, 1.3.1.99 is the PII for chassis 1, slot 3, port 1, (virtual) circuit ID 99. While a PII sometimes resembles an IP address, PIIs are totally different and should not be confused with IP addresses.

cmIface Objects

Each cmIface entry contains several objects. Three of the objects, Name, GroupName, and AdditionalInfo, are configurable identification for operator convenience. You can assign arbitrary strings to the interface Name and AdditionalInfo objects. Also, you can create interface groups and optionally assign each interface to a single group. These objects have no effect on how the CE150 processes user data. They are for customized interface identification only.

All other cmIface objects configure or display various aspects of how the CE150 processes and forwards user data. Not all objects are meaningful in all contexts. A typical cmIface entry for a DSL port might look similar to this example:

```
CRAFT> get cmiface [1.4.1]
Group: cmIfaceTable
Instance:             = [1.4.1.0]
PII                   = 1.4.1.0
IfIndex               = 1.4.1.0
Name                  = ""
GroupName             = ""
AdditionalInfo        = ""
NetModel              = Cross-Connect
IpAddr                = 0.0.0.0
NetMask               = 0.0.0.0
MacAddr               = ff.ff.ff.ff.ff.ff
BurnedInMacAddr       = 0.0.0.0.0.0
FarEndAddr            = 0.0.0.0
DestPII               = 1.2.1.61
ICPCompatible         = Yes
EncapsulationType     = PPP-HDLC
FwdMode               = PPP-HDLC-1973
```

Some objects are operator-configurable and others are read-only. Read-only objects display information about the interface; you cannot set or change the values. A general description of each object follows, but typically they may have different significance for different interface types (DSL, DSL VC, Ethernet, WAN VC), and in different netmodels.

IfIndex (Read/Write)	The PII of the interface. You cannot set this object, but you must specify it to identify the cmIface entry of interest. You can create and delete cmIface entries.
Name (Read/Write)	An arbitrary string that names the interface.
GroupName (Read/Write)	The name of the interface group, if any, to which this interface belongs.
AdditionalInfo (Read/Write)	An arbitrary string of information for the interface. For example, the name and phone number of a person to contact when there is a problem with this interface.
NetModel (Read/Write)	The netmodel that the CE150 uses for this interface. A NetModel of None means the interface is not configured for user data.

IpAddr (Read/Write)	In the IP netmodel, the standard IP address of this interface
NetMask (Read/Write)	In the IP netmodel, the standard IP subnet mask for this interface
MacAddr (Read-only)	In some netmodels, the working MAC address of this interface.
BurnedInMacAddr (Read-only)	For CPEs that support this feature, the burned-in MAC address of the CPE at the end of the DSL link.
FarEndAddr (Read/Write)	For IP netmodel WAN VCs, the IP address of the device at the opposite end of the WAN VC.
DestPII (Read/Write)	For most netmodels on DSL ports, the aggregated interface to which this DSL port is connected.
ICPCompatible (Read/Write)	For DSL ports, specifies whether the CPE at the end of the DSL link supports Copper Mountain's Internal Control Protocol (ICP) for enhanced network manageability.
EncapsulationType (Read/Write)	The data format of data frames or packets exchanged on this interface.
FwdMode (Read-only)	The kind of processing that the CE150 will perform for packets on this interface. The CE150 displays this result as a function of all other configuration.

Automatic Creation of WAN VC cmIface Entries

For ease of configuration file storing and loading, the CE150 allows the full set of cmIface entries to specified in any order. This means that you can specify a WAN VC as a DestPii of a DSL port, even if the WAN VC's cmIface entry does not yet exist. In this case, the CE150 automatically creates a minimal cmIface entry for the WAN VC. The CE150 sets the WAN VC's netmodel to the netmodel of the DSL port which implicitly created it. The CE150 sets all other parameters to the simplest defaults. In general, you should explicitly configure your WAN VCs to be sure all the parameters are set as you expect.

IP Netmodel

This section assumes you are familiar with the networking concepts of IP, and the CE150's IP capabilities. It provides details of how to configure the IP functions of the CE150. The IP netmodel is supported on DSL ports, Ethernet ports, and WAN VC interfaces.

Full IP Routing

DSL and Ethernet interfaces

The CE150 treats DSL interfaces the same as Ethernet interfaces, because in IP netmodel, the DSL link acts as a long cable to the CPE's Ethernet interface. The CE150 requires you to configure IP subnets on DSL and Ethernet interfaces. You must specify the NetModel, IpAddr, NetMask, and Encapsulation objects. For example:

```
CRAFT> craft> set cmiface[1.4.1] netmodel=ip ipaddr=10.0.0.1
netmask=255.0.0.0 encapsulationtype=rfc1483
```

WAN VC interfaces

The CE150 treats WAN VCs as unnumbered IP point-to-point links. Unnumbered links have no subnet on them, and each end may have an IP address. You can assign addresses to either end of the VC, both ends, or neither end. To assign an IP address to the CE150's end of the VC, set the IpAddr object. The NetMask is ignored for WAN VCs, because there is no subnet associated with a WAN VC. To assign an IP address to the far end of a WAN VC, set the farEndAddr object. Note that for the CE150 to route a packet to a WAN VC, one end or the other of the VC must have an IP address, because the routing table requires an IP address for the NextHop object (specified through the IPRoute MIB group). For example:

```
CRAFT> set cmiface[1.2.1.61] netmodel=ip ipaddr=10.0.0.1
farendaddr=192.168.1.1
```

Policy-to-WAN-VC IP Routing

You can configure DSL interfaces for policy IP routing to a WAN VC. To do so, set the NetModel, IpAddr, NetMask, Encapsulation, and DestPii objects in the DSL port's cmIface entry. You must specify a WAN VC as the DestPii. This means that inbound packets that are not addressed to the DSL's own IP address are always forwarded to the specified DestPii. For policy routing, the CE150 does not consult any routing table to forward received packets. For example:

```
CRAFT> set cmiface[1.4.1] netmodel=ip ipaddr=192.168.1.1
netmask=255.255.255.0 encapsulationtype=rfc1483
destpii=1.2.1.61
```

Ethernet and WAN VCs do not support policy routing.

Policy-to-Ethernet IP Routing

You can configure DSL interfaces for policy IP routing to the CE150's Ethernet port. To do so, set the NetModel, IpAddr, NetMask, Encapsulation, DestPii, and farEndAddr objects in the DSL port's cmIface entry. You must set the DestPii to a CE150 Ethernet port, usually 1.3.1. You must also set the DSL port's farEndAddr object to the next hop IP address on the Ethernet subnet to receive the forwarded packets. Each DSL port can specify a different farEndAddr on the same Ethernet interface. In this forwarding mode, the CE150 forwards inbound DSL packets that are not addressed to the DSL's own IP address, to the specified Ethernet port and next hop IP address. For policy routing, the CE150 does not consult any routing table to forward received packets. For example:

```
CRAFT> set cmiface[1.4.1] netmodel=ip ipaddr=192.168.1.1
netmask=255.255.255.0 encapsulationtype=rfc1483
destpii=1.3.1 farendaddr=172.16.1.1
```

Ethernet and WAN VCs do not support policy routing.

VWAN Netmodel

This section assumes you are familiar with the networking concepts of Copper Mountain's VWAN netmodel. It provides details of how to configure the VWAN functions of the CE150.

VWAN is the simplest mode to configure for DSL interfaces. To do so, set the NetModel, Encapsulation, and DestPii. You must specify a WAN VC or an Ethernet port for the DestPii. If only one DSL port refers to a given WAN VC, both interfaces are in VWAN point-to-point mode. If two or more DSL ports refer to the same WAN VC, all the interfaces in the VWAN group are in VWAN bridge mode. The CE150 automatically selects the FwdMode based on the given configuration. For example:

```
CRAFT> set cmiface[1.4.1] netmodel=vwan
encapsulationtype=rfc1490 destpii=1.2.1.61
```

WAN VCs do not need to be configured for VWAN mode because they assume VWAN mode automatically when referenced by VWAN DSL ports.

WAN VC and Ethernet interfaces cannot be configured as members of a WAN group.

Cross-Connect Netmodel

This section assumes you are familiar with the networking concepts of Copper Mountain's Cross-Connect netmodel. It provides details of how to configure the Cross-Connect functions of the CE150.

Ethernet interfaces don't support the Cross-Connect netmodel.

WAN VCs do not need to be configured for Cross-Connect mode because they assume Cross-Connect mode automatically when referenced by DSL ports.

VC to VC Frame Forwarding

In this mode, the CE150 treats the physical DSL interface as a Frame Relay interface. To configure this mode, set the DSL port's NetModel, and set Encapsulation=Q922. For each VC on the DSL interface, configure a cmSubIface entry with the DSL VC's PII, and set the cmSubIface DestPii to the WAN VC to which the DSL VC is Cross-Connected. For example:

```
CRAFT> set cmiface[1.4.1] netmodel=cross-connect
encapsulationtype=q922

CRAFT> set cmsubiface[1.4.1.40] destpii=1.2.1.61
```

HDLC to VC Frame Forwarding

In this mode, the CE150 forwards HDLC frame payloads into WAN VC payloads. To configure this mode, set the DSL port's NetModel, Encapsulation=HDLC, and DestPii. You must specify a WAN VC as the DestPii. For example:

```
CRAFT> set cmiface[1.4.1] netmodel=cross-connect
encapsulationtype=hdlc destpii=1.2.1.61
```

PPP Frame Conversion

In this mode, the CE150 converts PPP-HDLC frames on the DSL link to PPP-RFC1973 format on a WAN VC. To configure this mode, set the DSL port's NetModel, Encapsulation=PPP-HDLC, and DestPii. You must specify a WAN VC as the DestPii. For example:

```
CRAFT> set cmiface[1.4.1] netmodel=cross-connect
encapsulationtype=ppp-hdlc destpii=1.2.1.61

CRAFT> set cmiface[1.2.1.16] netmodel=cross-connect
encapsulationtype=rfc1973
```

CopperVPN Netmodel

This section assumes you are familiar with the networking concepts of Copper Mountain's CopperVPN netmodel. It provides details of how to configure the CopperVPN functions of the CE150.

DSL interfaces

To configure CopperVPN mode, specify the NetModel, Encapsulation, and DestPii. (The IpAddr is not meaningful in this mode.) You must specify a WAN VC as the DestPii. For example:

```
CRAFT> set cmiface[1.4.1] netmodel=coppervpn
encapsulationtype=rfc1490 destpii=1.2.1.61
```

Note that DSL ports support either encapsulation mode, RFC-1483 or RFC-1490.

WAN VC interfaces

Specify the NetModel and the farEndAddr. The farEndAddr is the IP address of the servicing router at the far end of the WAN VC. If you set the farEndAddr to non-zero, this is the IP address that the CE150 uses on the DSL interfaces. For example:

```
CRAFT> set cmiface[1.2.1.61] netmodel=coppervpn
farendaddr=172.16.1.1
```

If you set the farEndAddr to zero, the CE150 uses Inverse-ARP to learn the IP address of the servicing router. You can read this learned address be getting the WAN VC's cmIface entry. The farEndAddr will be returned as the learned address, and the FwdMode will be CopperVPN-auto, to indicate that the farEndAddr is learned and not operator-configured. For example:

```
CRAFT> set cmiface[1.2.1.61] netmodel=coppervpn farendaddr=0
```

HDIA Netmodel

With this netmodel, the CE150 is able to provide economical IP addressing for hosts. You can map all data VCs on different DSL ports to a single data VC on a WAN port. You can also map all voice VCs on different DSL ports to a single voice VC on a WAN port. HDIA, unlike VWAN or CopperVPN in their aggregate mode, permits peer-to-peer communications between DSL VCs.

Specify base IP addresses and address ranges (or net masks) for voice and data subnets when you are configuring the circuits on WAN ports. Also specify base IP addresses and address ranges (or net masks) for IADs and hosts when you are configuring circuits on DSL ports. For more detailed instructions, see *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual.

Here are examples of the configurations for the data and voice VCs on both the DSL and WAN interfaces.

```
CRAFT> set cmiface [1.2.1.20] netmodel=hdia ipaddr=10.0.20.0
netmask=255.255.255.192 encapsulationtype=rfc1483
farendaddr=10.0.20.1

CRAFT> set cmiface [1.2.1.21] netmodel=hdia ipaddr=10.0.21.0
netmask=255.255.255.192 encapsulationtype=rfc1483
farendaddr=10.0.21.1

CRAFT> set cmiface [1.5.13] netmodel=hdia ipaddr=10.0.20.2
netmask=255.255.255.255 destpii=1.2.1.20
encapsulationtype=rfc1483

CRAFT> set cmiface [1.5.13.22] netmodel=hdia
ipaddr=10.0.21.2 netmask=255.255.255.255 destpii=1.2.1.21
encapsulationtype=ip-1490
```

Appendix C
Event and Alarm Reference

This appendix contains an alphabetic reference listing of all CE150 events, divided into subsections for Alarms, Alarm-Clearing Notifications, and Event Notifications.

For descriptions of the groups and objects, see the *CopperEdge 150 CopperCraft Reference and MIB Definitions* manual.

Alarms

Type	Severity	Description
AtmVccDown	Minor	Sent by the System Control Module when an ATM WAN VC OperState causes changes to one of the fault conditions. *Information included in the event:* PII of the VC VPI of the VC VCI of the VC Operational status of the VC Cause of the operational status *Cleared by:* AtmVccUp *Trap:* Enterprise
BoardDown	Major	Sent by the System Control Module when a board stops responding to polls. *Information included in the event:* Index *Cleared by:* BoardUp *Trap:* cmBoardDown
BootFileFail	Major	Sent when a file download request from a board is not successful. *Information included in the event:* File name Board PII Reason code Descriptive text *Cleared by:* BootFileSucceed *Trap:* cmBootFileError
ConfigReadFailed	Major	Sent by the System Control Module when an error is encountered while reading the Config file. *Information included in the event:* File name Reason *Cleared by:* ConfigReadSucceed *Trap:* cmConfigFileReadFailed
ConfigWriteFailed	Major	Sent by the System Control Module when an error is encountered while writing the Config file. *Information included in the event:* File name Reason *Cleared by:* ConfigWriteSucceed *Trap:* cmConfigFileWriteFailed
CpePlugAndPlayFailure	Minor	Sent by the System Control Module when a CPE Plug and Play update fails. *Information included in the event:* Status string *Cleared by:* CpePlugAndPlayClear *Trap:* cmCpePlugAndPlayFailure

Type	Severity	Description
DLCIStateDisabled	Minor	Sent by the System Control Module when a Frame Relay PVC becomes disabled. *Information included in the event:* IF index DLCI State Port name Port PII Group name *Cleared by:* DLCIStateEnabled *Trap:* frDLCIStatusChange
DS1LineStatusAlarm	Major	Sent by a Quad T1 Wan module or a T1 line module when a line status changes to an error status. *Information included in the event:* Value of the DS1 line status Time at last line status change IF index of the DS1 line Port name Port PII Group name *Cleared by:* DS1LineStatusClear *Trap:* dsx1LineStatusChange
DS3LineStatusAlarm	Major	Sent by a DS3 module when a line status changes to an error status. *Information included in the event:* Value of the DS3 line status Time at the last line status change IF index of the DS3 line Port name Port PII Group name *Cleared by:* DS3LineStatusClear *Trap:* dsx3LineStatusChange
EndPointConflictAlarm	Minor	Sent by System Control Module if there is a bundle configure conflict; the end points of the IMUX bundles are improperly configured or connected. *Information included in the event:* PII of port that just trained EndPoint ID reported by port Logical IMUX PII Status string *Cleared by:* Manually *Trap:* cmEndPointConflictAlarm
FanFault	Major	Sent by the System Control Module when a Fan fault is detected. *Information included in the event:* None *Cleared by:* FanFaultClear *Trap:* cmFanFailure

Type	Severity	Description
IDSLTimingLossAlarm	Major	Sent by an IDSL line module when the network timing is lost. *Information included in the event:* Board PII Configured timing mode *Cleared by:* IDSLTimingLossClear *Trap:* cmIDSLBoardTimingLoss
LinkDown	Minor	Sent by the System Control Module when a DSL link, Frame Relay link, or ATM link transitions to disabled from the enabled state. *Information included in the event:* Port index Port name Port PII Group name *Cleared by:* LinkUp *Trap:* linkDown
LoginSaturated	Warning	Sent by the System Control Module when an operator login is attempted and the number of active sessions is already at the maximum. *Information included in the event:* Operator IP Reason *Cleared by:* LoginAvailable *Trap:* cmLoginsSaturated
LoginSuspended	Warning	Sent by System Control Module when three successive login attempts for a specific user are unsuccessful. This is cleared when the suspended user logs in successfully. *Information included in the event:* Operator IP Reason *Cleared by:* LoginAllowed *Trap:* cmLoginsSuspended
MaintFailed	Minor	Sent by System Control Module when a maintenance command fails. *Information included in the event:* Maintenance command attempted Status of command attempted Reason code Status string *Cleared by:* MaintSucceeded *Trap:* cmMaintCmdStatusChange

Type	Severity	Description
PortMisprovisioned	Minor	Sent by the System Control Module for a DSL port when the line trains but is not configured; that is, the operational state of the port is Up, but the port is not configured. This event alerts the operator that an end user is trying to connect on a port that is not configured. *Information included in the event:* Status string *Cleared by:* Manually *Trap:* cmPortMisprovisioned
PowerSupplyFault	Major	Sent by the System Control Module when a Power Supply fault is detected. *Information included in the event:* None *Cleared by:* PowerSupplyClear *Trap:* cmPowerSupplyFailure
RateFallbackAlarm	Minor	Sent by the DSL line module when the link trains at rate lower than the configured data rate. *Information included in the event:* Port PII Current data rate *Cleared by:* RateFallbackClear *Trap:* cmRateFallback
RedundancyChanged	Minor	Sent by the System Control Module when a change in the redundancy state (Enable/disable) is detected. *Information included in the event:* SCM board PII Redundancy is enabled or disabled *Cleared by:* Manually *Trap:* cmRedundancyChange
RedundancyConflict	Minor	Sent by the Secondary System Control Module when a secondary module thinks it should be locked as primary, but the other side is already locked. Both modules are trying to assume the Primary role. *Information included in the event:* Index *Cleared by:* Manually *Trap:* cmRedundancyConflict
RoleChanged	Major	Sent by the System Control Module when its redundancy role (Primary/Secondary) changes. *Information included in the event:* SCM board PII New primary SCM board PII Failover was automatic or manual *Cleared by:* Manually *Trap:* cmRoleChange

Alarm-Clearing Notifications

Type	Severity	Description
AtmVccUp	Notification	Sent by the System Control Module when an ATM WAN VC OperState changes to Up or enabled. *Information included in the event:* PII of the VC VPI of the VC VCI of the VC Operational status of the VC Cause of the operational status *Cleared by:* AtmVccDown *Trap:* Enterprise
BoardUp	Notification	Sent by the System Control Module when a new module is detected in the system. *Information included in the event:* Index *Cleared by:* BoardDown *Trap:* cmBoardUp
BootFileSucceed	Notification	Sent when a file download request to a module is successfully carried out. *Information included in the event:* File name Board PII Reason code Descriptive text *Cleared by:* BootFileFail *Trap:* cmBootFileOK
ConfigReadSucceed	Notification	Sent by the System Control Module when the Config file is successfully read. *Information included in the event:* File name *Cleared by:* ConfigReadFailed *Trap:* cmConfigFileReadOk
ConfigWriteSucceed	Notification	Sent by the System Control Module when the config file is successfully written. *Information included in the event:* File name *Cleared by:* ConfigWriteFailed *Trap:* cmConfigFileWriteOk
CpePlugAndPlayClear	Notification	Sent by the System Control Module when a CPE Plug and Play failure has been cleared. *Information included in the event:* Status string *Cleared by:* CpePlugAndPlayFailure *Trap:* cmCpePlugAndPlayClear

Type	Severity	Description
DLCIStateEnabled	Notification	Sent by the System Control Module when a Frame Relay PVC becomes enabled. *Information included in the event:* IF index DLCI State Port name Port PII Group name *Cleared by:* DLCIStateDisabled *Trap:* frDLCIStatusChange
DS1LineStatusClear	Notification	Sent by a Quad T1 Wan module or a T1 line module when line status changes to a no-error status. *Information included in the event:* Value of DS1 line status Time at last line status change IF index of DS1 line Port name Port PII Group name *Cleared by:* DS1LineStatusAlarm *Trap:* dsx1LineStatusChange
DS3LineStatusClear	Notification	Sent by a DS3 module when line status changes to a no-error status. *Information included in the event:* Value of DS3 line status Time at last line status change IF index of DS3 line Port name Port PII Group name *Cleared by:* DS3LineStatusAlarm *Trap:* dsx3LineStatusChange
FanFaultClear	Notification	Sent by the System Control Module when a Fan fault is cleared. *Information included in the event:* None *Cleared by:* FanFault *Trap:* cmFanFailureClear
IDSLTimingLossClear	Notification	Sent by an IDSL line module when the network timing is reacquired. *Information included in the event:* Board PII Configured timing mode *Cleared by:* IDSLTimingLossAlarm *Trap:* cmIDSLBoardTimingLoss

Type	Severity	Description
LinkUp	Notification	Sent by the System Control Module when a DSL link, Frame Relay Link, or ATM link transitions to enabled from the disabled state. *Information included in the event:* Port index Port name Port PII Group name *Cleared by:* LinkDown *Trap:* linkUp
LoginAllowed	Notification	Sent by the System Control Module when a successful login ends a login suspension for a particular operator. *Information included in the event:* Operator IP Reason *Cleared by:* LoginSuspended *Trap:* cmLoginsAvailable
LoginAvailable	Notification	Sent by the System Control Module when a login session ends and causes an existing LoginsSaturated condition to clear. *Information included in the event:* Operator IP Reason *Cleared by:* LoginSaturated *Trap:* cmLoginsAvailable
MaintSucceeded	Notification	Sent by the System Control Module when a maintenance command succeeds. *Information included in the event:* Maintenance command attempted Status of command attempted Reason code Status string *Cleared by:* MaintFailed *Trap:* cmMaintCmdStatusChange
PowerSupplyClear	Notification	Sent by the System Control Module when a Power Supply fault clears. *Information included in the event:* None *Cleared by:* PowerSupplyFault *Trap:* cmPowerSupplyFailure
RateFallbackClear	Notification	Sent by a DSL line module when a link is not trained. *Information included in the event:* Port PII Current data rate *Cleared by:* RateFallbackAlarm *Trap:* cmRateFallbackClear

Event Notifications

Type	Severity	Description
atmfVccChange	Minor	The attributes of the ATM WAN VC have been modified. *Information included in the event:* IP of the VC VPI of the VC VCI of the VC *Cleared by:* None *Trap:* atmfVccChange
AttributeChanged	Notification	Sent by the System Control Module when a Set command completes successfully (a group, instance, or object has changed value). There is one event per attribute changed. *Information included in the event:* Operator name Operator IP Operator context String with Group, Instance, Object, Value *Cleared by:* None *Trap:* cmAttributeChange
BoardRestart	Notification	Sent by any module before a software initiated reset due to the reason supplied. *Information included in the event:* Reason code Descriptive text *Cleared by:* None *Trap:* cmBoardRestart
ColdStart	Notification	Sent by the System Control Module when it powers up. *Information included in the event:* None *Cleared by:* None *Trap:* coldStart
Diagnostic	Notification	Sent by any module when a miscellaneous unexpected event occurs that the operator should know about. *Information included in the event:* Text *Cleared by:* None *Trap:* cmDiagnostic

Type	Severity	Description
FallingThresholdAlert	Notification	Sent by the System Control Module when a performance value falls below its threshold. *Information included in the event:* Index Threshold variable Sampling method Actual value Threshold value to compare against Port name Port PII Group name *Cleared by:* None *Trap:* cmFallingAlarm
LoginFailed	Notification	Sent by the System Control Module when an attempted operator login fails; the authentication is invalid or incorrect. *Information included in the event:* Operator IP (Telnet sessions only) *Cleared by:* None *Trap:* cmLoginError
LoginSucceeded	Notification	Sent by the System Control Module when an attempted operator login succeeds; the authentication is valid. *Information included in the event:* Operator IP (Telnet sessions only) *Cleared by:* None *Trap:* cmLoginOK
Logout	Notification	Sent by the System Control Module when an operator logs out. *Information included in the event:* Operator IP Reason *Cleared by:* None *Trap:* cmLogout
LoopStatusChange	Notification	Sent by the System Control Module at the start and end of a loopback test. *Information included in the event:* Status of loopback test Unique identifier for this test PII of interface being tested Type of test being performed Person or entity running the test If Status=Done, the termination code If Status=Done, the total elapsed time of the test If Status=Done, the total transmit count for the test If Status=Done, the total received errors for the test Test sub type Additional error information Port name Port PII Group name *Cleared by:* None *Trap:* cmLoopStatusChange

Type	Severity	Description
RisingThresholdAlert	Notification	Sent by the System Control Module when a performance value exceeds its threshold. *Information included in the event:* Index Threshold variable Sampling method Actual value Threshold value to compare against Port name Port PII Group name *Cleared by:* None *Trap:* cmRisingAlarm

Appendix D
Upgrading System Software

This appendix provides procedures for installing upgrades to the CE150 software. Installation of a software upgrade is, by definition, a service-affecting event and should only be performed during scheduled maintenance at off-peak times.

If you receive notice of a software upgrade to the CE150, be sure to read any printed materials or files that accompany the new software. Those specific instructions will always be more important; the procedures presented here are a more generalized view of the sequence required to update the CE150 file system.

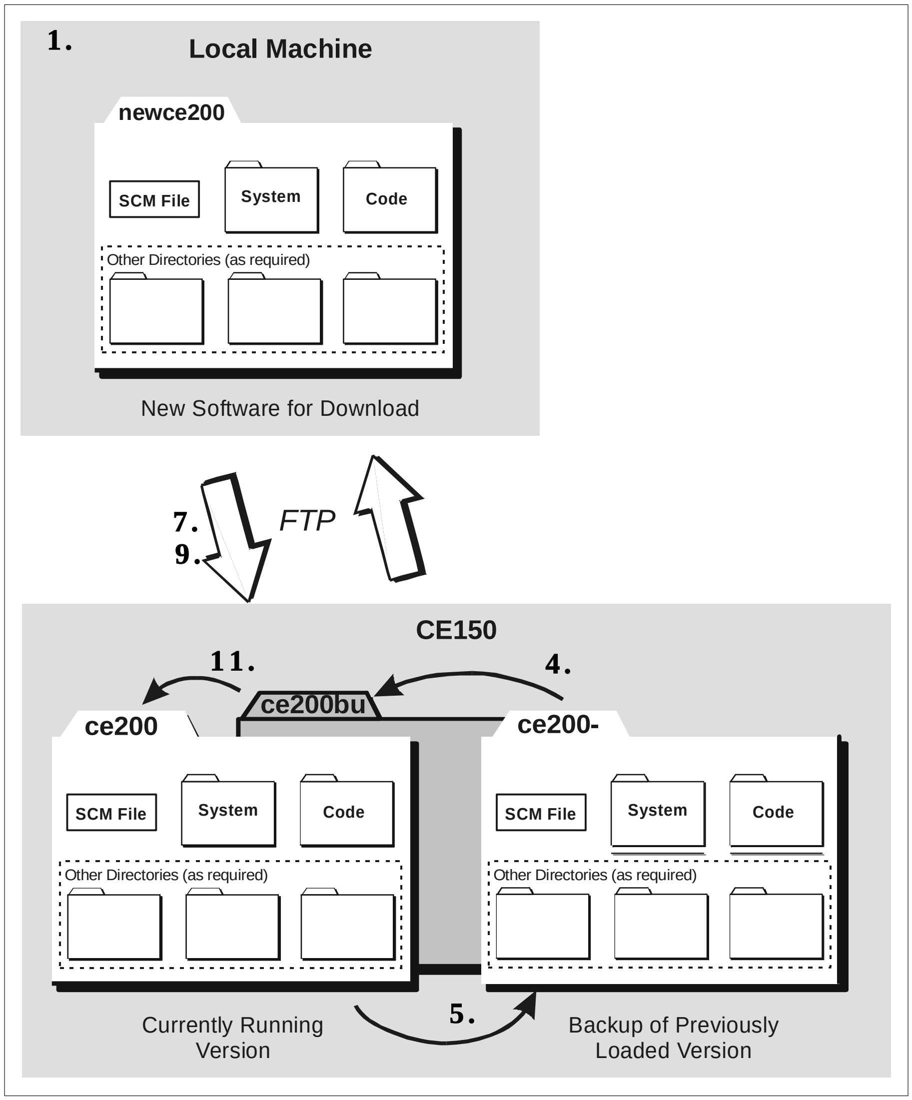
In the procedures, the term *local machine* refers to the computer (such as a PC) that you are using to control and manage the installation of new/upgraded software. We assume that your local machine is provisioned with suitable FTP and Telnet applications and that you are familiar with their use.

N O T E

Although you are upgrading a CE150 unit, the names of the directories are ce200 and ce200-.

Software Upgrade Overview

While specific software upgrades may have unique aspects, the basic tasks and their sequence include the following steps, as illustrated in the following diagram. The numbers in the diagram correspond to the numbered step.



Directory Structure for Software Upgrade

1. On the local machine, prepare a mirror of the CE150 directories containing the new software.
2. Telnet to the CE150 and save its current configuration (this updates the Config file).
3. Exit the Telnet session and then FTP to the CE150.
4. Rename the backup directory to deactivate it (rename ce200- to ce200bu).

5. Rename the current directory to make it the new backup directory (rename ce200 to ce200-).

6. Set the FTP utility to Binary mode and make sure the Prompt mode is turned on.

   If the screen prompt indicates that you have uploaded a file in ASCII mode, Binary mode has not been turned on. Delete the ASCII-transferred file to the CE150, and upload the file in Binary mode.

   By default, the FTP Prompt feature is turned on. Because of the switching among multiple directories on the local machines and the CE150s, complexity is built into the upgrade process. Keep the FTP Prompt feature active as a safety measure.

7. In the ce200bu directory, delete the System Control Module (SCM) file, and then upload (FTP *put*) the new SCM file from the local machine to the CE150.

8. In each subdirectory in the ce200bu directory, delete the files and then upload (FTP *put*) the corresponding new files from the local machine to the CE150.

9. Restore the system configuration by copying the Config file from the ce200- directory (FTP *get*) to the ce200bu directory (FTP *put*).

10. Rename the ce200bu directory (which now contains your new software) to ce200, thus making it the default code base when the CE150 starts up.

11. Restart the CE150 to activate the new software.

12. If the upgrade applies to the CopperRocket CPE as well as to the CE150, run the upgrade command (cmCPEBoard group) for each connected CPE unit. (See *Upgrade the CPE Software* on page 261.)

N O T E

*The compressed configuration file (config.tgz) is **not** compatible with any version of CE150 software prior to Version 3.0.*

FTP Errors

If you quickly perform several FTP operations, you may see a transient error from the CE200 because it has run out of sockets for an operation. Transient errors are numbered from 400 to 499, as in the following example:

```
ftp> dir
200 Port set okay
426 Data Connection error
```

Wait a few seconds and try the operation again. Under normal conditions, the problem will clear within 60 seconds.

Upgrade the CE150 Software

This section provides the generic procedures for upgrading the system software. If there is a conflict between these procedures and the specific instructions accompanying the new version of your software, follow the version-specific instructions.

Be sure to perform these procedures at the local machine (such as a PC) from where the new software will be copied.

Prepare the New Software on the Local Machine

1. On the local machine, create a directory (such as *newce200*).
2. Copy the new software to the newce200 directory.

   The new software is located at the Copper Mountain ftp site; you can access it directly through ftp or through the Copper Mountain web site. If you do not know how to access the ftp site, contact the Copper Mountain Technical Assistance Center.

   Although individual upgrades may vary, your new directory should contain at least the following file and directories:

   scm—a file with the new System Control Module code

   code—a directory containing the software files

   system—a directory containing the Config file

   These items are identical to the contents of the ce200 directory on the CE150.
3. Set the local machine's default directory to newce200.

N O T E

If the file or directory structure of the new software is not the same as shown here, refer to the instructions/release notes accompanying the new software for specific instructions.

Save the Current Config File on the CE150

1. Use Telnet or an SNMP Manager to connect to the CE150.
2. Log in to the CE150.
3. Enter **`admin`** when you are asked for the login name and password.
4. At the prompt, save the current configuration:

```
CRAFT> set cmsystem command=save
```

5. Verify that the save was successful:

```
CRAFT> get cmsystem
Group: cmSystem
ObjectClass         =  System
OperState           =  Enabled
Version             =  E 3.0
Master              =  0.0.0.0
ConfigFileName      =  config.tgz
CalendarTime        =  2000/07/21-10:01:11
MyPII               =  1.2.0.0
PrimaryPII          =  1.2.0.0
SecondaryPII        =  0.0.0.0
Redundancy          =  NotAvailable
ShelfCount          =  1
ExpIpSubNet         =  192.168.250.0
ConfigSynch         =  Saved
Command             =  SaveConfig
CommandStatus       =  Succeeded
```

6. If the CommandStatus object displays `InProgress` instead of `Succeeded`, repeat the **`get cmsystem`** command.

   If it displays `Failed`, contact Copper Mountain Technical Support.

7. Log out of the CE150.

Upgrade the Software on the CE150

Rename the Current and Backup Directories

1. Initiate an FTP session to the CE150 and log in.

```
C:\>ftp DSLAM_IpAddress
Connected to DSLAM_IpAddress
User <DSLAM_IpAddress:<none>>
```

2. Enter **`admin`** when you are asked for the user name and password.

3. Check the current directory structure:

```
ftp> dir
200 Port set okay
150 Opening ASCII mode data connection
  size          date       time       name
--------       ------     ------    --------
     512    Nov-09-2099  12:08:08   CE200      <DIR>
     512    Nov-09-2099  12:08:08   CE200-     <DIR>

    1024 bytes.  1248768 bytes free.
226 Transfer complete
252 bytes received in 0.00 seconds <252000.00 Kbytes/
sec>
```

   Note that two directories are listed with similar names: ce200 and ce200- ("ce200 minus"). The ce200- directory contains the code and supporting files for the last version of the software that ran on this CE150 *before* the version currently installed.

   If the CE150 is unable to boot from its current software (contained in the ce200 directory), it automatically looks for the ce200- directory and tries to boot using the previous software version.

4. Rename the ce200- directory to ce200bu. This version of the software is now obsolete and is about to be replaced:

```
ftp> rename ce200- ce200bu
350 Accepted source filename. Ready for destination
name
354 Rename done.
```

5. Rename the ce200 directory to ce200-. The current code is now designated as the backup:

```
ftp> rename ce200 ce200-
350 Accepted source filename. Ready for destination
name
354 Rename done.
```

Upload New Files

1. On the CE150, change to the ce200bu directory:

```
ftp> cd ce200bu
250 Changed directory to "P:/ce200bu"
```

2. List the ce200bu directory:

```
ftp> dir
200 Port set okay
150 Opening ASCII mode data connection
  size          date        time       name
--------       ------      ------    --------
     512    Jan-01-2001  12:08:08   .          <DIR>
     512    Jan-01-2001  12:08:08   ..         <DIR>
     512    Jan-01-2001  12:08:08   code       <DIR>
     512    Jan-01-2001  15:01:32   system     <DIR>
 2287440    Jan-29-2001  12:08:08   scm

 2289488 bytes.  56043520 bytes free.
226 Transfer complete
431 bytes received in 0.01 seconds <43.10 Kbytes/sec>
```

N O T E

You will only be upgrading the code and system directories. Ignore any other directories (such as html, image, or other).

3. Set up the FTP utility:

   a) Set the FTP utility to interactive mode:

   ```
   ftp> prompt
   Interactive mode On.
   ```

   If the following message is displayed, enter **prompt** again to turn Interactive mode on:

   ```
   Interactive mode Off.
   ```

   b) Set the FTP utility to binary code transfer mode:

   ```
   ftp> bin
   200 Type set to I, binary mode
   ```

   c) Set the FTP utility to view the progress of your uploads:

   ```
   ftp> hash
   Hash mark printing On <2048 bytes/hash mark>
   ```

4. On the local machine, change to the new software directory:

```
ftp> lcd c:\newce200
Local directory now C:\newce200
```

5. On the CE150, delete the SCM file in the ce200bu directory:

```
ftp> del scm
250 Command accepted.
```

6. Upload the new SCM file from the local machine to the CE150:

```
ftp> put scm
200 Port set okay
150 Opening BINARY mode data connection
####################################################
####################################################
####################################################
####################################################
####################################################
####################################################
################
226 Transfer complete
2254784 bytes sent in 50.94 seconds <44.26 Kbytes/
sec>
```

N O T E

When transferring software using FTP, a certain amount of delay is normal. Do not interrupt the process and try to start over—the original file on which the processor is running has been deleted from flash. Note also that while upload of the SCM file is in progress, the CopperCraft command line interface will not respond to commands. Wait until the SCM upload is complete before you enter further commands. If upload of the SCM file has not completed after 20 minutes, contact Copper Mountain's Technical Assistance Center.

7. Upgrade the Code subdirectory:

   a) Change to the Code subdirectory on both the CE150 and local machines:

   ```
   ftp> cd code
   250 Changed directory to "P:/ce200bu/code"

   ftp> lcd code
   Local directory now C:\newce200\code
   ```

   b) On the CE150, delete all *.bgz or *.bin files, one by one:

   ```
   ftp> del bc.bgz  or  del bc.bin
   250 Command accepted
   ```

   c) Upload the new code files (*.bgz) from the local machine to the CE150:

   ```
   ftp> mput *.bgz
   mput bc.bgz?
   ```

   Enter **y** for each file to confirm the upload.

   ```
   200 Port set okay
   150 Opening BINARY mode data connection
   ####
   ```

```
226 Transfer complete
14509 bytes sent in 0.10 seconds <145.09 Kbytes/
sec>
```

8. Upgrade the System subdirectory:

   a) *On the local machine only*, change to the System subdirectory:

   ```
   ftp> lcd ../system
   Local directory now C:\newce200\system
   ```

   b) On the CE150, change to the System subdirectory in the ce200- directory. It contains the Config file that you saved at the beginning of this process:

   ```
   ftp> cd /ce200-/system
   250 Changed directory to "P:/ce200-/system"
   ```

   c) Copy the Config file:

   ```
   ftp> get config.tgz  or  get config.txt
   200 Port set okay
   150 Opening BINARY mode data connection
   #
   226 Transfer complete
   3218 bytes received in 0.17 seconds <18.93
   Kbytes/sec>
   ```

   d) Return to the System subdirectory in the ce200bu directory:

   ```
   ftp> cd /ce200bu/system
   250 Changed directory to "P:/ce200bu/system"
   ```

   e) Delete the existing Config file in the ce200bu directory:

   ```
   ftp> del config.tgz  or  del config.txt
   250 Command accepted.
   ```

   f) Paste the Config file that you copied in step c:

   ```
   ftp> put config.tgz  or  put config.txt
   200 Port set okay
   150 Opening BINARY mode data connection
   #
   226 Transfer complete
   3218 bytes sent in 0.17 seconds <18.93 Kbytes/sec
   ```

9. At this point, the ce200bu directory on the CE150 contains all of the files needed for your upgrade.

10. Return to the system root directory:

    ```
    ftp> cd /
    250 Changed directory to "P:/"
    ```

Rename the Directory CE150 the New Software

From the CE150 root directory, rename the newly rebuilt ce200bu directory to the default name for the CE150 software (ce200):

```
ftp> rename ce200bu ce200
350 Accepted source filename. Ready for destination name
354 Rename done.
```

Restart the CE150 to Implement the New Software

1. Exit the FTP session:

```
ftp> bye
```

2. Use Telnet or the serial (Craft) interface to connect to the CE150.
3. Log in to the CE150, using **admin** for the login name and password.
4. At the prompt, restart the system:

   ```
   CRAFT> set cmsystem command=restart
   Set Successful
   ```

   It may take several minutes for the system to restart. During this time, the command line interface will not accept commands

   A good indicator that enough time has elapsed for the restart sequence to complete is when connection to the CE150 is lost.
5. When the system has completed its restart sequence, log into the CE150, using **admin** for the login name and password.
6. Verify that the new code is running on all System Control, Buffer Control, WAN, and DSL modules:

   ```
   CRAFT> geta cmboard swver
   SwVersion

   Instance: [1.2.0.0]
   5.0.83

   Instance: [1.3.0.0]
   5.0.83

   Instance: [1.4.0.0]
   5.0.83

   Instance: [1.5.0.0]
   5.0.83

   Instance: [1.6.0.0]
   5.0.83

   Instance: [1.7.0.0]
   5.0.83

   Instance: [1.8.0.0]
   5.0.83
   ```

   Check that the displayed software version matches the version of your upgrade.

View the Compressed Configuration File

There may be occasions when you want to view the Configuration file (config.tgz), which has been compressed using the gzip utility.

If you do not have the gzip utility on your PC, you can download it from the following web site:

http://www.gzip.org

Follow these steps to view the Configuration file:

1. Through FTP, manually upload the **config.tgz** file to your PC.
2. At the DOS prompt, rename the **config.tgz** file to **config.txt.gz**.
3. Run the gzip utility to unzip the file:

```
gzip -d config.txt.gz
```

The file is unzipped and named config.txt. You can view it in any text editor.

W A R N I N G

Do not ever update your config files with a text editor.

Upgrade the CPE Software

If an upgrade affects the operation of CopperRocket CPEs attached to this CE150, the CPEs may also need to have their software updated.

CopperRockets, as well as some third-party CPEs (consult the manufacturer's documentation for each equipment type), may be upgraded by downloading software from the CE150 individually using cmCPEBoard Upgrade command, or in groups using cmEndPointConfig.

For CPEs with unique capabilities (IMUX/Multilink or IAD devices used with derived voice applications), use the special instructions in *CPE Upgrades from an External Server* on page 262.

Upgrading the CPE software is service-affecting for the CPE being upgraded. During the upgrade process, communication with the CPE will be interrupted while it reboots, downloads and installs its new code, and then reboots from the new software. Should the CPE fail to train using the new code, the CPE will reboot and request a new download.

Upgrading an Individual CPE

Depending on the PROM version installed, some CopperRocket 201 CPEs do not have the 1.568 Mbps data rate hard-coded in their CPE PROM.

If you are performing upgrades of individual CR201s, be sure to perform the entire procedure below. To upgrade individual CPEs *other than* CR201, however, simply issue the upgrade command shown in step 2.

1. Set the SDSL Module Port to any data rate other than 1.568Mbps:

   CopperEdge> **set cmhdslmodem [*pii*] datarate=784**

2. Issue the upgrade command to the CPE:

   CopperEdge> **set cmcpeboard [cpe:*pii*] command=upgrade**

3. When the upgrade is complete, set the SDSL Module Data Rate to any supported rate.

Downloading software to a CopperRocket typically takes about two minutes to complete.

Upgrading Multiple CPEs

To upgrade or otherwise download software to multiple CPEs, it is generally more efficient to designate DSL physical ports that are eligible for bulk CPE download with cmEndPointConfig, and then use the cmMaintCmd group to execute a BulkDownload on all eligible CPEs at the same time.

Note that any necessary speed adjustments are performed automatically and the procedure described in *Upgrading an Individual CPE* is not necessary.

See the *CopperEdge 150 Installation and Operating Guide* for information on bulk downloading software to multiple CPEs.

CPE Upgrades from an External Server

To upgrade the software on connected IMUX CPEs or VoDSL IADs, perform the following steps:

1. Upgrade the software in the CE150 flash.
2. Restart the CE150 unit (System Restart).
3. From the *CPE* subdirectory of the *Release* directory on the Copper Mountain Support FTP site, **get** the following files from the subdirectory and transfer them to the host machine to be used as your remote file server. Be sure that the server IP address is reachable from the CE150's route table.

   cpe5_b.bgz(code for CopperRocket 202 IMUX SDSL)

   cpe5_h.bgz(code for CopperRocket 212 IMUX IDSL)

   cpe5_t.bgz(code for CopperRocket 408 IAD)

N O T E

These code files (identified by the .bgz file extension) are contained within a compressed archive. The CopperEdge will automatically expand the files when installing them on the CPE, so do not take any specific action except as specified below.

4. When the system has fully initialized and all connected CPEs have trained, log into the CE150 and configure the cmFile group as follows:

FSAddr:	The IP Address of the file server.
FSUsername:	The User Name used to log in to the file server
FSPass:	The Password used to log in to the file server.

FSDir: The directory path to be traversed to reach the file archives you transferred in step 3.

5. Once the CPE code files have been placed on the remote server, and the cmFile group has been configured to locate them, follow the procedure in *Upgrading an Individual CPE* on page 261 (in this release, Bulk Upgrade is not supported for IAD or IMUX CPE).

   When you issue the **upgrade** command, the CE150 will identify the CPE as a type whose code is stored on the remote server. The files will then be copied to the CE150, expanded, and downloaded to the specified CPE.

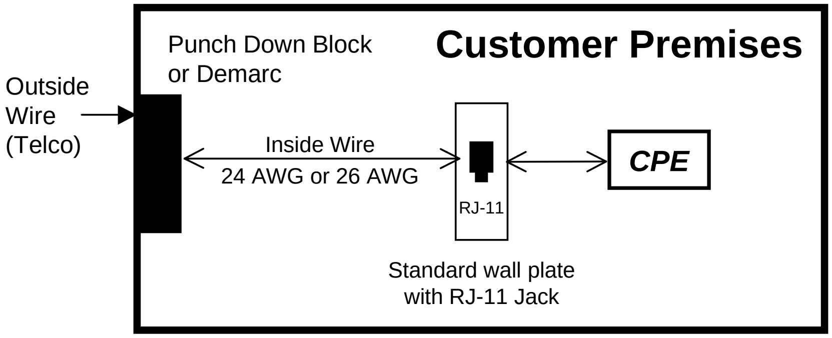
   For IMUX CPEs, you only need to enter the PII of *one* of the DSL physical interfaces that comprise the IMUX link.

Appendix E
CPE Inside Wiring

Inside wiring for Copper Mountain's CopperRocket xDSL family of CPEs consists of a standard, single-pair telephone facility to which the DSL line is connected, and to which the CPE itself connects with a standard 2- or 4-conductor (RJ-11/RJ-14) cable. A diagram is included on the next page for your reference.

If you are using a third-party CPE manufactured by someone other than Copper Mountain Networks, check the manufacturer's documentation for any special requirements.

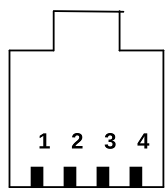
Inside Wiring for xDSL CPE with Copper Mountain Networks COE



Inside Wiring Specifications:

Pinouts

Pin 1 = Yellow
Pin 2 = Green
Pin 3 = Red
Pin 4 = Black



Connect CPE using standard RJ-11/RJ-14 telephone cable included with the CPE.

Only Pins 2 and Pin 3 are used; connection is straight through with no crossover. Unit automatically adjusts for Tip and Ring on these pins.

Recommend inside wire: 24 AWG or 26 AWG standard telephone wire; CAT 3 and CAT 5 UTP can also be used.

Appendix F
Glossary

This glossary is provided as an aid to recognizing some of the specialized terms used in this manual that relate to the Copper Mountain DSL system, the CE150 and the Copper-View EMS. Selected telephony and data communications terms commonly used in DSL networking are also included.

A

ADSL Asymmetrical DSL. DSL scheme in which upstream and downstream data rates are different, typically with the downstream channel running much faster than the upstream channel.

address A character string used as a unique identifier, usually of a specific location, user, machine, or logical name.

agent An active object with a specific purpose. *See* SNMP Agent.

AIS Alarm Indication Signal used in DS1 and DS3 status messages.

alarm An event category that requires an operator's attention.

ARP Address Resolution Protocol. A protocol within the TCP/IP suite that maps IP addresses to Ethernet addresses. TCP/IP requires ARP for use with Ethernet.

ATM Asynchronous Transfer Mode. A typically high-speed (T3+) method of wide-area data transfer in which the information is organized into cells. Asynchronous in the sense that there may not be any periodicity in the recurrence of information cells originating with a specific end point or user.

B

BER Bit Error Rate; also Block Error Rate.

boot, bootstrap The process of starting a computer or a routine, the first few instructions are sufficient to bring the rest of it's necessary components into memory.

BootP server An external device that provides bootstrap data through the Bootstrap Protocol.

bps (bits per second) Measurement unit used to specify the rate or speed of data communications.

Buffer Control Module A module used with the CE150 that manages the transfer of data between the proprietary interface to the DSL Modules, and between the PC interface and the System Control Module.

C

CCV C-bit Coding Violation Error (for C-bit Parity and SYNTRAN DS3 applications).

CE CopperEdge. The generic term that refers to all of the Copper Mountain DSLAM class of equipment. *See also* DSLAM.

CE150, CE200 CopperEdge 150 and CopperEdge 200 are high-density, multi-speed DSL systems (fast packet access concentrators), which allow network service providers to offer high-speed local access to the Internet and to private enterprise networks.

CES C-bit Errored Seconds. A second with one or more CCVs, one or more Out of Frame defects, or a detected incoming AIS.

Clear An event (CE-generated or manual) that causes alarm indications (visual and audible) to disappear. *See also* Event.

CLI Command Line Interface. *See* CopperCraft.

CMCP Copper Mountain Compatible Protocol.

COE Central Office Equipment. Communications apparatus that normally resides in a telco central office or with a Network Service Provider. The CopperEdge 200 is one type of COE.

cold boot The process of restarting a computer system as if it were powered off, then back on. Also called cold start.

complex Consists of a System Control Module (SCM), a Buffer Control Module (BCM), and optionally one or two WAN modules. *See also* Primary complex *and* Secondary complex.

control master The single SCM in a CE system that is the point of egress for all user data traffic to the WAN, and the point of operator management of the system.

control module The System Control Module (SCM) or Buffer Control Module (BCM).

CopperCraft Copper Mountain's product name for its embedded command-line software that utilizes an SNMP-like command language to configure and monitor performance of DSL and other ports on the CE150.

CopperVPN Copper Mountain's name for a networking model by which subscribers are aggregated into a virtual private network of DSL links, connecting transparently with a remote LAN or corporate network, but without reliance on any special capabilities or software in router devices connected to the CoppperVPN circuit.

CPE Customer Premise Equipment. Telecommunications equipment that resides at the subscriber's location. Copper Mountain's CopperRocket 201 SDSL Modem is one type of CPE.

D

datagram A message unit that contains source and destination address information, as well as the data itself, which is routed through a packet-switched network.

DCE Data Communication Equipment. In defining interface standards for connection of equipment to a network, the classification of a device as Data Communications Equipment (typically a modem or printer) or Data Terminal Equipment (typically a computer or remote terminal device) determines the wiring of the interface connector. In networking parlance, the network access device is normally considered the DCE.

DHCP server A host processor accessed through Dynamic Host Configuration Protocol.

DLCI Data Link Circuit Identifier. A logical name (generally 10 bits in length) assigned to a Frame Relay virtual circuit and included in the Frame Relay header.

DLCMI Digital Link Connection Management Interface. Software mechanism for configuring and controlling Frame Relay virtual circuits.

DSL Digital Subscriber Line. A digital telephone line capable of reliably transporting data at high speeds (for G.lite, from 160 Kbps up to 2 Mbps; for G.dmt, from 32 Kbps up to 6 Mbps) over ordinary twisted-pair copper wire local loops. Also, the hardware and software technology utilized to implement DSL.

DSLAM Digital Subscriber Line Access Multiplexor. A network device, usually at a telephone company central office, that receives signals from multiple customer DSL connections and sends the signals on a high-speed backbone line using multiplexing techniques. Depending on the product, DSLAM multiplexers connect DSL lines with some combination of asynchronous transfer mode (ATM), Frame Relay, or IP networks.

DSL Module A CE150 module that interfaces between DSL links and the switched network. CE150 DSL modules may either be 12- or 24-port types, and, depending on the model, may support a variety of xDSL formats.

DTE Data Terminal Equipment. In defining interface standards for connection of equipment to a network, the classification of a device as Data Communications Equipment (typically a modem or printer) or Data Terminal Equipment (typically a computer or remote terminal device) determines the wiring of the interface connector. Network interfaces on the CE150 (Ethernet and WAN) are wired as DTE.

E

Ethernet A local area network used for connecting computers and peripheral devices over short distances and using private facilities. Ethernet operates over twisted pair wire or coaxial cable at rates to 10 Mbps or 100 Mbps.

Event An indication of an occurrence that an operator may want to know about.

F

filter A logical construct that allows selective reception or blocking of data packets based on the criteria (such as the IP address of the originator) assigned and built into the filter specification.

Frame Relay A high-performance shared-bandwidth packet-switching protocol, which has become the de-facto standard in wide-area data communications.

FTP File Transfer Protocol.

H

HDIA High Density Internet Access. A networking model that allows separate voice and data circuits on DSL and WAN ports and that provides economical use of IP addressing.It requires that you set up base IP addresses and address ranges (or net masks) for voice and data subnets on WAN ports as well as on DSL ports for IADs and CPEs and their hosts.

HDSL High-rate Digital Subscriber Line. Originally referred to DSL links at T1 (1.544 Mbps) and E1 (2.048 Mbps) rates. This use of the term is becoming less common, however, as DSL rates have generally increased.

host A processor (computer) that provides services such as database access or computation to other devices. In commands used to configure a CE150, *host* may refer to an external file server or work station, or to the CE150 itself, depending upon the operational organization of the site.

I

IAD Integrated Access Device. A special type of CPE that provides dual circuits, accommodating digitized on one circuit and regular data on the other. On the upstream side, both circuits communicate with a CE150 over an SDSL line. On the downstream, for voice, POTS lines interface with the IAD; for data, an Ethernet line interfaces with the IAD.

IDSL DSL service in which the data rates are those used in ISDN services (64, 128, or 144 Kbps). IDSL services typically allow longer loops between the CPE and COE, and may allow use of existing ISDN CPE in the DSL system.

IMUX An acronym for *Inverse Multiplexing*, a technique by which data from a single source is separated into multiple streams for transmission over multiple physical links, effectively multiplying the throughput available to an individual subscriber.

interface A point of demarcation (may be physical or logical) between two devices where the electrical signals, connectors, timing and handshaking are defined. It can also refer to the procedures, codes, and protocols that enable a connection and exchange of information between two entities.

IP address Internet Protocol address. A 32-bit address assigned to hosts using TCP-IP. The address is written as four octets that denote the addressee's network, subnet, and host location.

IRB Integrated Router/Bridge. A special operating mode of certain routers that allows the router to support virtual bridge groups, and which thus enables the CE150 Virtual Wide Area Network (VWAN) capability.

ISDN Integrated Services Digital Network. A CCITT standard for a digital network as implemented and delivered by telecommunications common carriers.

L

LAN Local Area Network. A short distance network used to link computers and peripheral devices under standard control. LANs allow users to access centralized servers and anyone on the LAN can send messages to and work jointly with others on the LAN.

LCV Line Coding Violation (a class of DS3 error).

LMI Local Management Interface. A Frame Relay specification related to information exchange between network devices (bridges, routers, etc.).

LOS DS1 and DS3 Loss of signal message.

M

MAC address Media Access Control address. A unique 48-bit character string (address) permanently assigned to an individual device at the time of manufacture.

Mbps Megabits per second, used to denote a data rate. A rate of 1 Mbps is 1 x 10^6 bits per second.

MIB Management Information Base. A set of variables forming a database contained in an SNMP-managed node on a network. SNMP managers can fetch/store information from/to this database.

MIO Multiplexed Input/Output. A term applied to CE150 network interfaces.

multilink A technique in which data to or from a single endpoint is divided into multiple streams and transported over multiple physical links, effectively multiplying the transmission rate. *See* IMUX.

N

node A point where connection may be achieved into a network. In practical terms, a node is usually the site of a switch or router.

O

octet An eight-bit data byte. An IP address is comprised of four octets, each ranging in number from 0 to 255.

operational CE150 A CE150 that is powered up and passing data (user or management traffic).

P

PDU Protocol Data Unit. The standard abbreviation used to refer to SNMP protocol packets.

PES DS3 P-bit Errored Seconds. One second of time in which one or more P-bit coding violations, frame defects or incoming alarm signals occur.

PII Permanent Interface Identifier. A character string that provides a means of relating a *physical* interface location to its corresponding network identifiers (IP address, MAC address, etc.).

PIIs are formatted as c.ss.pp.uuuu. Where c is the chassis (shelf) number, ss is the slot number on the CE150 where the interface terminates, pp is the port number of the interface. If a virtual circuit is associated with the interface, uuuu is an identifying number for the virtual circuit, if applicable.

For example, the PII 1.3.2.27 refers to system shelf 1, the WAN card in Slot 3 of that shelf, port number 2, and a virtual circuit with a DLCI or ATM Virtual Link number of 27.

PPP Point-to-Point Protocol.

protocol A set of rules relating to the format and timing of data transmission between devices. In packet-switched networks, protocols break a lengthy message into equal parts called packets. The packets are transmitted across the network, the receiving processor then error-checks them, and returns an acknowledgment to the originating device.

protocol interface While a PII points to and focuses on a physical interface (a port or connector), a protocol interface is a logical name for any interface through which packet date passes: a physical port (DSL, Ethernet, etc.), or a virtual circuit. A protocol interface may take the form of a full PII (*always* including a circuit identifier), or it may simply be the logical name (cmName, IfName) assigned to that interface.

PSES DS3 P-bit Severely Errored Second. One second of time in which 44 or more P-bit coding violations, frame defects or incoming alarm signals occur.

PVC Permanent Virtual Circuit. A Frame Relay term for a configured connection between two devices, usually over a wide-area link. The “permanent” circuit will persist as long as its configuration remains in place. Once a PVC is configured between two devices, no setup or disconnect process is required to initiate or conclude communication over the channel.

R

RADIUS “Remote Authentication Dial-In User Service”, a protocol for carrying authentication, authorization, and configuration information between a device that requires authentication service and a shared server on which the authentication data resides.

runt frame Term used to describe an invalid or incomplete Ethernet data frame.

S

SDSL Symmetrical DSL. DSL service in which data rates are the same in both directions.

shelf A single CE150 unit, consisting of a chassis and all of its circuit modules.

SNMP Simple Network Management Protocol. A protocol used to monitor and provision nodes and devices across a network.

SNMP agent Device-based software used to monitor and report status when queried by an SNMP manager. By definition, SNMP agents run on managed devices; an SNMP manager is required to perform active (command and configuration) functions.

SNMP manager System software used to remotely query and configure network devices. A more generic term, also used in this document, is Element Management System.

subnet address An extension of the Internet addressing scheme; allows a location to use a single IP address for multiple physical networks.

subnet mask A 32-bit address mask used in IP to specify a particular subnet.

System Control Module The CE150 circuit module that controls timing, protocol processing and overall system functionality.

T

TCA Threshold Crossing Alert. A condition which generates an alarm on the CE150. Using the cmAlarmTable which itself uses counters that measure conditions like overruns, underruns, and errors, you can tell the CE150 to indicate when the number of errors becomes unacceptable.

trap An unsolicited messaged generated by an SNMP agent on a network device due to a threshold value being exceeded or the occurrence of a pre-defined event.

TCP/IP Transmission Control Protocol/Internet Protocol. Currently the most widely used layered transmission protocol for connecting dissimilar computers across networks.

TDR Time Domain Reflectometer. A test device that acts on RADAR-like principles to determine the location of metallic faults.

Telco The Incumbent Local Exchange Carrier (ILEC) or the Competitive Local Exchange Carrier (CLEC).

TFTP Trivial file Transfer Protocol. A simplified version of FTP used to transfer files based on UDP (User Datagram Protocol) with no guarantee of delivery. TFTP lacks password protection and certain other capabilities of a full-featured FTP application.

U

UBR Unspecified Bit Rate. Used to describe certain ATM links.

V

VC Virtual Channel. An ATM communications channel; often used to mean any multiplexed packet stream aggregated into a virtual circuit.

VCC Virtual Channel Connection. A concatenation of ATM VCLs. *See* VCL.

VCI Virtual Channel Identifier. Numeric (16-bit) field in an ATM cell header that identifies the virtual channel the cell is to take.

VCL Virtual Channel Link. A discrete end-to-end ATM connection that encompasses both directions of the data exchange. Analogous to the Frame Relay PVC.

VP Virtual Path. A unidirectional ATM path consisting of a logical group or bundle of VCs.

VPI Virtual Path Identifier. A 8-bit field in a cell header designating the virtual path to be taken by the cell.

VPL Virtual Path Link. The full path between the point at which a VPI is assigned to a Cell and the point at which the VPI is translated or removed (i.e, between the starting line and finish line of the path).

VWAN Virtual Wide Area Network. A networking model that makes use of the IRB (Integrated Router/Bridge) capability of compatible routers to allow transparent connectivity of LAN ports over wide area (Frame Relay) links.

W

WAN Wide Area Network. WANs cover extended geographical areas in contrast to LANs. WANs generally use links provided by the public switched telephone network to connect distant sites.

Index

Symbols

Numerics

A

B

C

D

E

F

COPPER MOUNTAIN

Sheet ______ of _______

DSL Physical Port Configuration

Perm. Ifc. Index Chassis .	Slot .	Port	Subscriber IP Address	Account Number	Account Name	Install Date	Status / Notes
		1					
		2					
		3					
		4					
		5					
		6					
		7					
		8					
		9					
		10					
		11					
		12					
		13					
		14					
		15					
		16					
		17					
		18					
		19					
		20					
		21					
		22					
		23					
		24					

COPPER MOUNTAIN

Sheet ______ of _______

WAN Port Configuration Worksheet

Perm. Ifc. Index				Port IP Address	Subnet Mask	FWD Mode	Net Model	Encaps. Type	CIR	EIR	Interface/Link Notes
Shelf .	Slot .	Port .	VC No.								